WANDL

File Format Reference for NPAT and IP/MPLSView

Release

6.2.0

17 August 2015

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

WANDL File Format Reference for NPAT and IP/MPLSView

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Table of Contents

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes. Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/cgi-bin/docbugreport/.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/
- To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico). For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

Chapter 1

Introduction to IP/MPLSView File

Specification File and Default Parameters

File Type	Description	Chapter
spec	Specifies locations for the different files describing the network	Chapter 2, Specification File
dparam	Specifies hardware type and contains default values for network parameters	Chapter 3, Dparam File

Backbone Files

The notation bbdsgn refers to the text mode version of IP/MPLSView. It can be invoked from the command line, at $WANDL_HOME/bin/bbdsgn. When running in text mode, files can be created using the Modify Configuration Menu.

The muxloc and demand files should be created first. Afterwards, IP/MPLSView can perform a design from scratch to decide where backbone links should be bought. The backbone links can then be saved to a file and used as the bblink file for future runs. If a bblink file is supplied and specified in the specification file, IP/MPLSView performs simulation and design to decide where new backbone links are required in order to place all demands in the demand file(s), and to satisfy diversity constraints. A domainfile can also be created/modified by using the Modify Configuration Menu.

The contents of these files will not automatically be changed by the program after each design. New backbone configurations, path assignments, or changes made during the design can only be saved by the user. To use these new files, the specification file must reference them. These files are described in the following chapters.

The following list is a partial list of files.

Node Related Files

File Type	Description	Page
muxloc	Main file for defining nodes and their geographical coordinates	Muxloc File on page 29
graphcoord	Specifies graphical (as opposed to geographical) coordinates	Graphcoord File on page 28
nodeparam	Specifies hardware types of nodes	Nodeparam File on page 33
site	Specifies the site to which a node belongs	Site File on page 35
group	Specifies a logical node group	Group File on page 28
domain*	Specifies the domain to which a node belongs	Domain File on page 27
facility*	Specifies the facility to which a node belongs	Facility File on page 45
hpnni*	Specifies the peer group to which a node belongs (for ATM PNNI)	Separate manual
nodeweight	Control File	Node Constraints on page 83
vpn*	Specifies the VPNs that exist the network	VPN File (IP/MPLS only) on page 36
bgpnode*	Specifies the BGP speakers of the network	Bgpnode File (IP/MPLS Only) on page 37

* Files denoted by an asterisk are optional and license dependent.

Link Related Files

File Type	Description	Page
bblink	Main file for defining links by nodes and trunk types	Bblink File on page 39
bgplink*	Specifies the BGP neighbors of the network. (Router-specific)	Bgplink File (IP/MPLS Only) on page 43
facility*	Specifies the facility to which a node or link belongs	Facility File on page 45
fixlink	Control File	Fixing Links on page 83
linkdist	Control File	Setting the Administrative Weight on page 81
rsvbwfile	Control File	Reserved Bandwidth on page 84
intfmap	Interface mapping file	Interface File Outbound (IP/MPLS Only) on page 60 and Interface File Inbound (IP/MPLS Only) on page 61
policymap*	CoS Policy mapping file (Router-specific)	Policymap File (IP/MPLS Only) on page 48

* Files denoted by an asterisk are optional and license dependent.

Demand and Traffic Related Files

File Type	Description	Page
demand	Main file for defining network demands	Demand and Newdemand Files on page 51
newdemand	File for defining additional network demands	Demand and Newdemand Files on page 51
CoS*	Lists all classes of service and their priority (Router-specific)	CoSAlias File (IP/MPLS Only) on page 59
CoSAlias*	Lists all the classes of service aliases and the class of services that are mapped to them (Router-specific)	CoSAlias File (IP/MPLS Only) on page 59
interfaceLoad_in*	Defines the incoming traffic load for interfaces in 24-period intervals (Router-specific)	Interface File Inbound (IP/MPLS Only) on page 61
interfaceLoad_out*	Defines the outgoing traffic load for interfaces in 24-period intervals (Router-specific)	Interface File Outbound (IP/MPLS Only) on page 60
owner	Specifies the owner of a node or demand	Owner File on page 63
pathtable	Specifies the path tables at a node	Pathtable File on page 63
srvcprofile	Specifies a distribution of service type	Srvcprofile File on page 64
srvctype	File for defining demand types	Srvctype File on page 65
trafficload*	File for defining the peak load at different time intervals	Trafficload File on page 67
tunnel*	Main file for defining tunnels (Router-specific)	Tunnel (Tunnelfile) (IP/MPLS Only) on page 71
trafficpattern*	Defines traffic characteristics	Trafficpattern File on page 74
trafficdata*	Defines permanent virtual circuits	Trafficpattern File on page 74

* Files denoted by an asterisk are optional and license dependent.

Chapter 2

Spec File

Spec File

Description

A specification file is accessed by the IP/MPLSView commands to determine which directories, parameters file, and input files to use.

Syntax

runcode=*fileextension*

Substitute *fileextension* with a file extension that you would like to use by default for your network files when they are saved.

datadir=*directory*

This directory is the default location that bbdsgn will look into for input files specified in the specification file. The datadir directory '.'. Here, a period, without quotes, indicates the containing directory of the specification file. In older versions, a period represented the working directory. However, to simplify things, this usage was changed.

ratedir=*directory*

This directory contains rate tariffs, such as private line and voice tariff rates. The default ratedir directory is /u/wandl/db/rates/default. If customized tariffs are used, ratedir should point to the directory containing the customized tariffs.

filetype=filepath

Substitute *filetype* with the specification file type (e.g., muxloc, bblink, demand, dparam, etc.). Substitute *filepath* with the name of the input file if it is in the datadir, or the absolute path for the file if it is not in the datadir. For a filetype not in use, either leave out the entry for that filetype or substitute filepath with "none".

Informational Note: The word "none" is reserved to indicate that no file is associated/specified.

Usage

Each entry should be on a separate line.

Adding Comments

In the specification file and in input files (dparam, muxloc, etc.), comments can be indicated with a pound sign or a pound sign followed by an exclamation mark. Comments marked in the former way (with ‘#’) are ignored by the program and no attempt is made to preserve them when saving a network. Comments marked in the latter way (with ‘#!’) are preserved by the program when saving a network if they are on the same line as a critical line, e.g., at the end of a valid link entry.

Example

In the sample specification file shown below, named spec.sept, some of the input filenames have a .sept extension.

```
 runcode = sept
 datadir = .
 ratedir = /u/wandl/db/rates/default

# The following files are in the datadir directory

 dparam = dparam # parameter file

 # backbone data
 muxloc = muxloc.sept  # Mux node file
 nodeparam = nodeparam.sept # mux type and constraints file
 site = site.sept  # site definition file
 domainfile = none  # domain name and color definitions
 demand = dpath.sept  # circuit demands/paths specification
 newdemand = none# second demand file
 bblink = bblink.sept  # backbone configuration
 facility = facility.sept     # facility file

 # cost files
 ratespec = none  # year-term/volume discount specification
 usercost = none  # designate the user-defined cost
 bbfacility = none  # describe existing hardware facilities

 # optional control files
 rsvbwfile = none # defines reserved bandwidth for specific node pairs
 nodeweight = none
 fixlink = none
 graphcoord = none  # user defined node positions

 # device specific/IPMPLS
 tunnelfile = tunnel.sept # tunnel definition file
 tunnelbitfile = tbit.sept # tunnel bit file
 t_trafficload = traffic.sept # tunnel traffic file
```

Chapter 3

DPARAM File

This chapter explains the dparam file.

Dparam File

Description

The dparam file is used to indicate hardware related parameters, link bandwidth and overhead parameters, size and performance tuning parameters, and miscellaneous parameters.

Syntax

Parameters in Alphabetical Order

Parameter	Category	Page
addroute2treename	VLAN	addroute2treename on page 25
adjt3ndwt	designparam	adjt3ndwt on page 12
advFilterThreshold	miscellaneous	advFilterThreshold on page 25
batch	miscellaneous	batch on page 26
bbestpct	accessdesign	bbestpct on page 11
bbovhd	accessdesign	bbovhd on page 11
bbtype	designparam	bbtype on page 12
bumpflag	experimental	bumpflag on page 18
checkpir	routing	checkpir on page 19
CheckTransitDemandLimit	routing	CheckTransitDemandLimit on page 18
chk1link	diversitydesign	chk1link on page 16
chkalllink	designparam	checkalllink on page 12
chksitenode	diversitydesign	chksitenode on page 16
configloopaddrinpath	pathdesign	configloopaddrinpath on page 17
corebbtype	designparam	corebbtype on page 12
cos2lspmap	miscellaneous	cos2lspmap on page 26
currency	pricing	currency on page 20
custrate	pricing	custrate on page 20
divgrouplevel	pathdesign	divgrouplevel on page 18
divpathbw	pathdesign	divpathbwpct and divpathbw on page 18
divpathbwpct	pathdesign	divpathbwpct and divpathbw on page 18
dsgnNoPathSlct	designparam	dsgnNoPathSelect on page 12

Parameter	Category	Page
E1bw	linkbw_ovhdparam	Link Bandwidth and Overhead Parameters on page 24
E1lkovhd	linkbw_ovhdparam	
E3bw	linkbw_ovhdparam	
E3lkovhd	linkbw_ovhdparam	
ECMPcntByBW	ecmp	ECMP Parameters on page 25
estdsgncost	pricing	estdsgncost on page 20
estusercost	pricing	estusercost on page 20
extratrunkpenalty	designparam	extratrunkpenalty on page 12
fatpct	designparam	fixfat and fatpct on page 13
fixfat	designparam	fixfat and fatpct on page 13
forceCoreRoute	designparam	forceCoreRoute on page 13
forcedesign	designparam	forcedesign on page 14
framesize	routing	
frcL3simreroute	simulationoption	frcL3simreroute on page 22
frcprefval	designparam	frcprefval on page 14
fts2000	pricing	fts2000 on page 20
gatewayweight	routing	gatewayweight on page 19
government	pricing	government on page 21
hopdelay	designparam	hopdelay (or nodepenalty) on page 14
hopdist	designparam	hopdist on page 14
hwvendor	hwparam	hwvendor on page 23
ignoreintldb	pricing	ignoreintldb on page 21
ignoremultiprocess	routing	ignoremultiprocess on page 19
incSizingBW	sizing	incSizingBW on page 23
keepl2sptree	VLAN	keepl2sptree on page 25
lambdabw	miscellaneous	lambdabw on page 26
lincyear	pricing	lincyear on page 21
linkdistunit	designparam	linkdistunit on page 15
linkdvpri	diversitydesign	linkdvpri on page 16
linklatencyvalue	miscellaneous	linklatencyvalue on page 26
llvendor	pricing	llvendor on page 21
MCsimrptopt	simulationoption	MCsimrptopt on page 22
matchtunnelname	miscellaneous	matchtunnelname on page 26
maxcallsetup	simulationoption	maxcallsetup on page 22
maxdivlinkcheck	diversitydesign	maxdivlinkcheck on page 16
maxECMPcnt	ecmp	ECMP Parameters on page 25
maxhop	designparam	maxhop on page 15
maxinterlink	designparam	maxinterlink on page 15
maxintralink	designparam	maxintralink on page 15
maxlink	designparam	maxlink on page 15
maxlinkcheck	designparam	maxlinkcheck on page 15

Parameter	Category	Page
maxnodenumber	designparam	maxnodenumber on page 15
maxSizingBW	sizing	maxSizingBW on page 22
mediadiv	experimental	mediadiv on page 19
minECMPflowbw	ecmp	ECMP Parameters on page 25
minSizingBW	sizing	minSizingBW on page 22
mixdesign	designparam	mixdesign on page 15
mplsenable	mpls	mplsenable on page 23
netgroom_demandtype	netgroom	netgroom_demandtype on page 17
netgroom_maxbw	netgroom	netgroom_maxbw on page 17
netgroom_maxpriority	netgroom	netgroom_maxpriority on page 17
netgroom_minawgain	netgroom	netgroom_minawgain on page 17
netgroom_linkstatus	netgroom	netgroom_linkstatus on page 17
netgroom_releasebw	netgroom	netgroom_releasebw on page 17
netgroom_reportorder	netgroom	netgroom_reportorder on page 17
noBGP	BGP	noBGP on page 23
OC12bw	linkbw_ovhdparam	Link Bandwidth and Overhead Parameters on page 24
OC12lkovhd	linkbw_ovhdparam	
OC12lkovhd	linkbw_ovhdparam	
OC3bw	linkbw_ovhdparam	
OC3lkovhd	linkbw_ovhdparam	
OC48bw	linkbw_ovhdparam	
OC48bw	linkbw_ovhdparam	
OC48lkovhd	linkbw_ovhdparam	
OC48lkovhd	linkbw_ovhdparam	
phyhoplimit	hwparam	phyhoplimit on page 23
print_link_dist_in_DVSIM	reportoptions	print_link_dist_in_DVSIM on page 21
printname	reportoptions	printname on page 21
prIPaddr	reportoptions	prIPaddr on page 21
randomflag	pathdesign	randomflag on page 18
reversepri	hwparam	reversepri on page 23
reportBWunit	reportoptions	reportBWunit on page 22
reportstyle	reportoptions	reportstyle on page 22
routeorder	routing	routeorder on page 19
siminterval	simulationoption	siminterval on page 22
simmaxhop	simulationoption	simmaxhop on page 22
sitedvpri	diversitydesign	sitedvpri on page 17
sizing_growthconstant	sizing	sizing_growthconstant on page 23
sizing_growthmultiplier	sizing	sizing_growthmultiplier on page 23
sizing_resizeopt	sizing	sizing_resizeopt on page 23
skipcount	designparam	skipcount on page 16
skiplinkdsgn	designparam	skiplinkdsgn on page 16

Parameter	Category	Page
status_report_interval	designparam	status_report_interval on page 16
T1bw	linkbw_ovhdparam	Link Bandwidth and Overhead Parameters on page 24
T1lkovhd	linkbw_ovhdparam	
T2bw	linkbw_ovhdparam	
T2lkovhd	linkbw_ovhdparam	
T3bw	linkbw_ovhdparam	
T3lkovhd	linkbw_ovhdparam	
tunnelovhd	hwparam	tunnelovhd on page 24
uselinkname	reportoptions	uselinkname on page 22
usepreemptpri	path	usepreemptpri on page 20
vendor	pricing	vendor=vendor on page 21

Usage

The dparam file is not required in order to run bbdsgn if the program is licensed only for one hardware type. For users who have license to multiple vendor hardware types, the `hwvendor` field should be specified in the dparam file.

Example

A dparam file containing the default parameter values is listed below. must be set in the dparam file if they are different from their default values. Other parameters may be changed interactively in the bbdsgn program.

```
batch = 0  # 0: interactive job, 1: batch
keeporigpath = 1 # routes read in from Demand files are kept for comparison
# Tariff Options
vendor = least cost, # default vendor for inter-LATA circuits
llvendor = LEC # default vendor for intra-LATA circuits
fts2000 = 0 # 1=FTS2000, 2=LINCS, 3=LINCS+FTS2000
government = 0 # 1=government application
currency = DL# DL = U.S. Dollar
estusercost = 1   # 1=estimate link cost from user-defined cost
# Report Option
printname = 0  # 0=node number, 1=node name
uselinkname = 0  # 0=node number, 1=link name in path spec
# Design Parameters
bbtype = T1 # default link type used in design
maxlink = 500 # maximum number of links in the backbone
hopdelay = 100
maxhop = 8       # Design: maximum number hops in a path
hopdist = DISTANCE
linkdistunit= 10 # default link distance (used when hopdist=user defined)
fixfat =     0   # reserved bandwidth per link. unit=bit
fatpct = 0.000   # reserved bandwidth percentage per link.
randomflag  =   1  # 1: randomly distribute circuits to best path solutions
maxnodenumber = 250    # maximum node number
# Diversity Design Parameters
sitedvpri = 0 # site and link diversity priority
linkdvpri = 0 # link diversity priority
chk1link =  # 1= delete single line in link diversity check
usepreemptpri = 0 # 0=use priority, 1=use preempt priority
adjt3ndwt = 1# weight adjustment for T3 nodes
checkalllink = 1 # link deletion not restricted by BBtype
```

```
# Access Design Parameters
bbestpct = 1.00 # backbone cost estimation adjustment percentage
bboverhd = 50.00 # backbone per port termination cost
# Hardware Related Parameters:
hwvendor =
phyhoplimit = 12# physical link hop limit
reversepri = 0 # 0=smaller number has lower priority
# Link bandwidth and overhead parameters
# Size and performance tuning parameters
maxlink =  1500 # maximum number of links in the backbone
maxintralink= 3  # max intra-lata link considered during design
# Misc. parameters for what if studies
mediadiv = 1 # 1: check for media diversity
bumpflag = 1
siminterval=   20  # FR simulation time, unit=second
```

Parameters in More Detail

Access Design Parameters

bbestpct

Backbone cost estimation adjustment percentage

bbovhd

Backbone per port termination cost

Usage

These two parameters are used by the program to adjust the backbone cost while doing the access design.

To decide if a circuit is cost effective to ride on the backbone, the program compares the direct offnet cost versus the sum of the access cost and backbone cost. The backbone overhead cost, bboverhd, can be used to indicate fixed overhead and hardware cost. The backbone cost adjustment percentage, bbestpct, can be used to adjust backbone hardware cost to encourage or discourage the homing.

We'll use one example to illustrate the usage of these parameters. Let A and B be the end locations of a circuit, say C1. Let MuxA and MuxB be the closest MUX backbone node for A and B respectively.

Let's use:

- Cost(X,Y) = the leased line cost from X to Y
- TCost(X,Y) = the leased line cost from X to Y

Then the circuit C1 is cost effective to ride on the backbone if

- Cost(A,B) >= Cost(A,MuxA) + Cost(B,MuxB) + bboverhd +bbestpct * TCost(MuxA,MuxB) * bw/T1bw;

where bw is the bandwidth of the circuit C1 and T1bw is the bandwidth of a T1.

The adjustment factor can be used to encourage the program to put more or less circuits on the backbone.

If the path from an offnet location to its closest backbone node goes through a RCL DIP location, the cost between the location and the backbone node is set to the cost between the location and the DIP location. The cost of the microwave link between the location and the backbone node is set to 0.

If the path from an offnet location to its closest backbone node goes through a low level concentrator (HUB) location, the cost between the location and the backbone node is set to the cost between the location and the HUB plus prorated cost of the line from the HUB to the backbone node.

Backbone Design Options

The shortest path first (SPF) algorithm is used to place circuits on the backbone. There are several ways to define the length of a link. Some examples include: constant distance, actual mileage, administrative weight, OSPF, delay metric, and CDV metric. If constant distance is assigned to all links, the SPF algorithm becomes one of minimum hop, which is the algorithm implemented by most MUX hardware.

Note that the actual methods available for determining the length of a link will depend on the hardware model being used. For example, the CDV metric will only be visible to users of an ATM switch model. Similarly, the IGRP distance calculation method will only be visible to users of the Cisco router model.

adjt3ndwt

This option adjusts the weight given to T3 nodes, affecting their desirability during design.

bbtype

The parameter bbtype specifies the primary backbone link type. If there is not enough capacity in the backbone to place all the circuit requirements, bbdsgn buys links of bbtype to satisfy the requirements. The resize option from the design menu can then be used to adjust the size of the links. Only links of the type specified by bbtype are bought by the program.

checkalllink

After performing a design, bbdsgn prompts the user whether to remove potentially redundant links. If the user answers yes to this prompt, and checkalllink is set to 0, bbdsgn checks only links with hardware type equal to the bbtype. If checkalllink is set to 1, bbdsgn checks all link types.

corebbtype

Default link type used in the design of the core.

dsgnNoPathSelect

Default value is 0. If set to integer *n*, the program will change the order of the demand *n* times after the initial design. After each change, the program will run another design and buy more links if needed. This parameter can be used when the actual hardware switch does not support the path select option.

extratrunkpenalty

Penalty for buying additional trunks at nodes exceeding NNI capacity.

If the dparam forcedesign flag is set to 1 and if extratrunkpenalty>=100 and the nodeparam flag BLOCKTRUNK is off, then trunks are allowed to be added to nodes whose NNI_Bw limit has been exceeded. This extratrunkpenalty is added to the new trunk as cost (in US Dollars) to decrease the possibility of being used.

Default value=0.

fixfat and fatpct

To reserve part of the link bandwidth for future growth or other reasons, the user may use the parameters fixfat and fatpct to define the amount of bandwidth to be reserved on each link. The parameter fixfat allows you to indicate the reserved bandwidth amount using a specific number, while fatpct expresses the reserved bandwidth as a percentage of the total bandwidth. Both parameters may be used together. In situations where only one of them applies, the other parameter should be set to zero. The amount of bandwidth available for path assignments is determined by the following calculation:

- link bandwidth - link overhead - fixfat - (link bandwidth * fatpct)

Link overhead will vary with the hardware device being used.

The fixfat and fatpct parameters will be overridden by any contradicting information in the reserved bandwidth file (rsvbwfile), if specified.

Consider using fixfat and fatpct 1) to reserve bandwidth for future growth, 2) to reserve bandwidth as the hardware vendor requests, 3) to reserve bandwidth for potential circuit overhead, or 4) to reserve bandwidth to reduce cell drop ratio and delay

Because of the bandwidth allocation algorithm implemented, or bugs in the version of firmware, the hardware vendor may ask you to leave a portion of the bandwidth not used. If your overall link utilization is very high and you are using the bump feature heavily, you should consider reserving bandwidth at large node locations.

If you are emulating a hardware switch not currently supported by bbdsgn, or if you have not defined all circuit constraints such as pass through timing, or asynchronous and transparent signaling, these parameters may be used to reserve the bandwidth for per circuit overhead.

For ATM, frame relay, and router networks, the user may need to reserve at least 20% of the total bandwidth. Otherwise, the over-subscription nature of packet switching may cause cell drop and delay during heavy traffic.

forceCoreRoute

The forceCoreRoute parameter is used to turn on or off a special routing constraint involving the node hierarchical levels of “core” and “regular”. When this constraint is turned on, no core->regular-> core routing is permitted. There cannot be in any part of the path a core node going to another core node through regular node(s). Once you begin at or transit through a core node, you can visit any number of core nodes. However, if you then go from a core node to a regular node, you can visit regular nodes but no more core nodes in the rest of the path.

If we take ‘r’ to represent a regular node and ‘c’ a core nodes, we could explain the routing restrictions using regular expressions. If forceCoreRoute is turned on, only (r-)*(c-)*(r-)* routing is allowed. (c-)*(r-)*(c-)* routing is not allowed in any part of the path.

Examples: Suppose R1,R2,R3 are regular nodes and C1,C2,C3 are core nodes. If the constraint is on, the program allows paths like R1-R2, C1-C2, R1-C1-C2-C3-R3, C1-C2-R3, and R1-R2-C3. However, it disallows paths like C1-R2-R3-C3 or R1-C1-R2-C2-R3.

- If forceCoreRoute is set to 0, this constraint is never applied.
- If forceCoreRoute is set to 1, this constraint is applied to the *design* mode.
- If forceCoreRoute is set to 2, this constraint is applied to both *design* and *simulation* modes.

forcedesign

This parameter takes on value 0 or 1. The default value is 1.

- forcedesign=1: The program will keep doing design, even if it cannot buy trunks to route some of the demands. It will discard the unroutable demands and continue the design.
- forcedesign=0: The program will quit if it encounters a demand that it cannot route even if it buys trunks.

frcprefval

If set to 0, bbdsgn will ignore the media prefer flag during backbone design. (The flag, M*media_pref*, where *media_pref* is substituted by a string of capital letters representing preferred media types.

hopdelay (or nodepenalty)

The nodepenalty is a penalty for buying trunks at a node. It is used during backbone design to encourage the program to prefer the purchase of direct links. This parameter can be increased to try to reduce the number of parallel links in the network. It can also be decreased if the routing is using too many direct links rather than using available bandwidth in a less direct route.

The nodepenalty parameter also gets used during routing, but only in the special case where the routing method is set to *Actual_mileage*. In this case, a higher value will influence the path placement to select paths with fewer hops, since the length of a path will be calculated using the following formula: (length of links in miles) + (number of nodes * hopdelay). Note that during a simulation, hopdelay is set to 0.

hopdist

The parameter hopdist is used to define the default link distance calculation method. It can be set to any of the values specified above such as: Constant, Actual_Mileage, Adm_Weight, OSPF, Delay, CDV_Metric, or hardware specific. If it is set to hardware specific, hopdist will default to the actual method used by the hardware.

If the value of hopdist is set to Adm_Weight, then the contents of the file linkdist, referenced in the spec file, are used to set the distances of the links. (Previously this value was “User_Defined.” All previous references to User_Defined will still be understood by the program.)

linkdistunit

The value, linkdistunit is used to set the distance for links not explicitly defined in the linkdist. If linkdistunit is set to a positive number, say 10, then the distance for all the links not specified in the linkdist file are set to 10. If linkdistunit is set to -1, then the distance for all the links not specified in linkdist are set according to their actual distance. If set to any other negative number, the following formula is used:

- distance = (distance - linkdistunit+1)/(-linkdistunit)

For example, if linkdistunit is set to -30, then the distance of a 250 mile link is set to 9.

(250-(-30)+1)/(-30) = 9

maxhop

The path placement parameter, maxhop, is used to indicate the maximum number of hops allowed in a path. Paths created by the bbdsgn program will not exceed this limit.

maxinterlink

Max inter-lata link considered during design.

maxintralink

Max intra-lata link considered during design. The parameter maxintralink is used by the backbone design subroutine, and only affects link designs within the United States. It is used to limit the number of checks within a LATA in order to speed up the design process. The bbdsgn program checks for up to maxintralink consecutive intra-LATA links while deciding the optimal locations to place backbone links. The default value is set to 3. If network nodes are clustered within a few LATAs, the value of this parameter may need to be increased to 4, 5 or even 6. Note that this parameter is only used by the program in backbone design steps in deciding where to buy new links.

maxlink

The parameter maxlink is the maximum number of links supported in a design. The default value is 1500. If a design requires more than 1500 links, this parameter may need to be set to a larger value. When the number of links exceeds maxlink during the design process, the design is stopped. The parameter maxlink affects the amount of memory space allocated by the bbdsgn program.

maxlinkcheck

This parameter specifies the maximum number of links checked for deletion during a design.

maxnodenumber

The parameter maxnodenumber defines the maximum number of backbone nodes that may be added during the design process. The parameter maxnodenumber affects the amount of memory space allocated by the bbdsgn program.

mixdesign

Design tuning parameter; default value=0

skipcount

This is a heuristic parameter to reduce the number of times a reroute is performed during the design process. A design case with skipcount=20 will take much less time to run than a case with skipcount=0.   Recommended value is 0, 5, 10 and 20. You can use values from 1 to 50. Default value is 0.

skiplinkdsgn

When the setting is turned on, this parameter sets the diversity design to only check for node/site diversity and skips link diversity checking during the design process. When the value is 1, this setting is turned on. Default value is 0, which means the setting is turned off.

status_report_interval

The unit is seconds. If you specify status_report_interval=*n*, for some integer *n*, the design status will be reported every status_report_interval.

Diversity Design Parameters

chk1link

This parameter is a link diversity definition, used for diversity design. It can be manually set, or set using bbdsgn.

chk1link value	explanation
0	Survive node-pair connectivity failure
1	Survive single private line failure
2	Survive single facility failure

An edge is defined as a group of private lines connecting the same node pairs, and having the same vendor and backbone type. Private lines in the same edge have the tendency to go down at the same time. As such, all the private lines in the same edge are taken down during the link diversity check for single link failures.

The chk1link parameter is used to adjust the link failure definition. If set to the default value of 0, all the private lines in an edge are taken down for single link failure check. If set to 1, only one private line in the link is brought down.

chksitenode

This parameter is a site diversity definition option. The values are specified as follows:

chksitenode value	explanation
1	Survive single site and single node failure
0	Survive single site failure
-1	Survive single node failure

linkdvpri

The diversity design subroutine insures that there is enough bandwidth in the backbone to route paths with priorities greater than or equal to linkdvpri during any single link failure.

maxdivlinkcheck

This parameter specifies the maximum number of links checked for deletion during a diversity design.

resizetype

This specifies the type of resize candidates. By default, it is set to “tariff” (for the types in the tariff database). Other options are “Nx64k”.

sitedvpri

The diversity design subroutine insures that there is enough bandwidth in the backbone to route pass through paths with priorities greater than or equal to sitedvpri during any single link, node, or site failure. The priority specified by sitedvpri should be greater than or equal to that of linkdvpri.

Net Groom

These parameters correlate exactly to the options for network grooming.

netgroom_demandtype

Specifies the type of demand to optimize.

netgroom_maxbw

Specifies the maximum bandwidth of demands to optimize.

netgroom_maxpriority

Specifies maximum priority of demands to optimize.

netgroom_minawgain

Any demands that will not achieve a gain in admin weight larger than this value when optimized will not be optimized.

netgroom_linkstatus

Selects which type of links to use when optimizing demands.

netgroom_releasebw

Specifies whether or not the original path of the demand being optimized will be factored into the network when optimizing a demand.

netgroom_reportorder

Specifies how the demands are sorted when a report is generated on the optimized demands.

Path Design Options

configloopaddrinpath

IP/MPLSView supports specifying LSP tunnel configured paths based on interface IP addresses through **Modify > Elements > Tunnels, Path Config Options: “Add” “Config”**. Additionally, configured paths can be recalculated in LSP tunnel path design in **“Design > TE Tunnels > Path Design”**.

If configloopaddrinpath=1 is specified in the dparam file, LSP tunnels’ configured paths would be created with the nodes' loopback IP addresses rather than with interface IP addresses, when the above options, "add config" or "path design," are performed.

divgrouplevel

To specify the diversity group level (site or linkdiversity) desired in Routing demands and tunnels in the same diversity group or path/standby path pair, use the divgrouplevel keyword in the dparam file. Set divgrouplevel to 3 for site diversity and 2 for link diversity.

dparam line	Type of diversity desired
divgrouplevel=2	link diversity
divgrouplevel=3	site diversity (where a site can contain one or more nodes.)

The diversity group name of a demand can be specified in the demand file under Type_field.

divpathbwpct and divpathbw

If you specify a bandwidth for a hot standby, that bandwidth will be used. Otherwise, bbdsgn will specify it for you based on user-specified, or else default, bandwidth parameters. You can specify for the bandwidth of the hot standby to be a given percentage bandwidth of the original demand/tunnel (divpathbwpct) plus an overhead (divpathbw) in the dparam file.

BW2=BW1*divpathbwpct+divpathbw

Where BW1=bandwidth of a demand/tunnel and BW2=the bandwidth of its hot standby.

The default values for divpathbwpct and divpathbw, if none are set in the dparam file, are 1 (for 100%) and 0.

randomflag

This option randomly distributes demands when calculating best path solutions, as opposed to using the sequential order of circuits in the input file.

Path Placement Options

bumpflag

For experimental studies. If there is not enough bandwidth in the backbone, some hardware devices will grab bandwidth from lower priority circuits and allocate it to higher priority circuits according to the priority and preemption assigned to each circuit. Due to timing constraints, most hardware grabs bandwidth from lower priority circuits while checking circuit paths for high priority circuits. This might cause unnecessary circuit reconnection overhead.

The bumpflag parameter is introduced as an experimental parameter. It is used to study the impact of failures on circuit path placement when the bump feature is disabled. When set to 0, the bump feature is disabled by the program. That is, the preemption field of the circuit path requirement is ignored by the path placement subroutine. The program assumes that high priority paths cannot bump low priority paths during path placement and simulation.

The default value of bumpflag is 1.

CheckTransitDemandLimit

If set to 1, node transit statistics will be displayed in the Node Traffic Summary Report.

If set to 2, group transit statistics will be displayed in the Group Traffic Summary Report and node transit statistics will be displayed in the Node Traffic Summary Report.

checkpir

Check the PIR during path placement

gatewayweight

A penalty, gatewayweight, is used to discourage the program from Routing the paths through too many domains. The default gatewayweight is 1000. This value cannot be changed interactively. To modify this value, add the following line to the dparam file:

```
gatewayweight=number
```

substituting *number* with a numeric value. For example, to change the gatewayweight value to 2000, add the following line:

```
gatewayweight=2000
```

ignoremultiprocess

To turn on multiprocess checking for OSPF routing instance analyses, set this parameter to 0. For more information, refer to the Router Guide chapter, “Routing Instances.”

mediadiv

For experimental studies. The default value of mediadiv is 1. In this case, the media diversity constraints specified in the circuit requirements are followed by the path placement and backbone design modules. If mediadiv is set to 0, the media diversity constraints defined in the circuit requirements are not followed.

routeorder

The route order is the Placement Order and can be set through the Design Options,Path Placement tab.

value	explanation
HPRI_INPUT	High priority first, input order
HPRI_RAND	High priority first, scramble
RAND	Scramble randomly
LOWBW_RAND	Low bandwidth first, scramble
DIST_RAND	Shortest distance first, scramble
INPUT	Input sequence
NODE_INPUT	Input sequence, scramble
HIBW_RAND	High bandwidth first, scramble
HPRI_HIBW_RAND	High priority first, then high bandwidth, scramble
HPRI_HIBW_INPUT	High priority first, then high bandwidth, inputorder

usepreemptpri

value	explanation
0 (default)	use priority
1	in the linkdvpri and the sitedvpri interpretation described above, use the prempt priorities (instead of the call priorities) of the circuits

Priority is a value assigned in the definition of a demand that determines the order in which it is placed during Routing. Preempt priority determines whether a demand is able to "bump" a circuit with a lower priority. In supported device models, the preempt priority actually maps to holding priority. Holding priority determines whether another circuit can bump the current one.

Pricing Options

currency

The currency parameter is assigned by a two letter code denoting the currency of a particular country. This parameter specifies the currency in which pricing information is displayed.

The default value of currency is DL, representing U.S. dollars.

custrate

If a link cannot be priced based on COST information in the bblink file or the user cost file, then bbdsgn can check the database that has the commercial tariff rates. A user can also define their own rates in the custrate file which can then replace the database. If custrate is set to 1, the program will use the custrate file. If set to 0, the program will use the database.

The default value, if the variable is not set in the parameter file, is 0.

estdsgncost

- 1: bbdsgn will estimate link costs for design purpose when link costs are not known.
- 0: bbdsgn will not buy links where pricing failed

estusercost

bbdsgn prices links by first searching through user-defined costs, and then through the default tariff database if the first search failed. The link cost can also be specified in the bblink file. If the cost is specified in the bblink file by the user, then that value will override the user-defined cost value in the usercost file. If no matches are found, the pricing for that link fails. However, the user has the option of allowing bbdsgn to estimate the cost of a link (for design purposes) by setting the option estusercost to 1. bbdsgn will subsequently look in the user cost information to determine an estimated cost. If no base can be found, bbdsgn will fail to price that link. bbdsgn will otherwise estimate service costs based on a similar service. As an example, if the tariff for a 256Kbps line is not available, but the price for a 512Kbps line is found, bbdsgn will estimate the cost of that 256K line based upon the cost of the 512K line.

fts2000

- fts2000 = 0: the interstate IXC tariffs and LEC special access tariffs are used to price out links and circuits.
- fts2000 = 1: AT&T FTS2000 tariff
- fts2000 = 2: MCI LINCS tariff

- fts2000 = 3: FTS2000 and LINCS tariffs

The LINCS tariff is offered to the FAA (Federal Aviation Administration) by MCI. Only users doing network design for the FAA will see this option in their bbdsgn version. If fts2000 is set to 3 and the IXC vendor is set to least cost, the vendor will be selected based on the lower of AT&T FTS2000 and MCI LINCS tariffs. Note that Sprint FTS2000 tariff is not yet supported.

The default value of fts2000 is 0.

government

Certain IXC POP are designated as GOVERNMENT ONLY switches. If the value of government is set to 0, switches marked GOVERNMENT ONLY will not be used when pricing private-line circuits. This parameter should be set to 1 for government networks, and 0 for all other networks.

ignoreintldb

1: bbdsgn will not use the international database for tariff information when links cannot be priced from the bblink file (COST=) or user cost file.

lincyear

This option is a customized tariff feature for U.S. government users.

llvendor

The parameter, llvendor, is used to define the default intra-LATA vendor. It can be set to least cost, LEC (for local exchange carrier), or same as default IXC vendor (ATT, USS, MCI, WTG). If the value of llvendor is set to same as default IXC vendor, it should have the same value as vendor. When different IXC vendors are set for these two parameters, bbdsgn changes the value of llvendor to the value specified for vendor.

The default value for llvendor is LEC.

vendor=*vendor*

The parameter, vendor, is used to define the default inter-exchange carrier (IXC) vendor. It can be set to ATT, USS (Sprint), MCI, WTG (WorldCom), or “least cost”.

The default value of vendor is “least cost”.

Report Options

printname

If printname is set to 0, node numbers or IDs are used to identify backbone nodes. If printname is set to 1, then the names specified in the muxloc file are used to identify backbone nodes in the reports. The user should first make sure that names in the muxloc file are unique if the printname parameter is to be set to 1.

print_link_dist_in_DVSIM

If print_link_dist_in_DVSIM is set to 1, the DVSIM and L2_DVSIM failure simulation reports will include the Geo_Dist column. Geo_Dist is the geographical distance between two nodes.

prIPaddr

If set to 1, link IP addresses will be used for path specification instead of linknames (assuming that uselinkname is also set to 1 in the dparam file).

reportBWunit

This parameter configures the bandwidth unit used in NPAT reports. Configurable values: b, Kb, Mb, Gb, Best.

reportstyle

Configurable values: CSV, TEXT (for formatted text), HTML

uselinkname

If uselinkname is set to 0, the node IDs (origination/termination) that define the link are used. If uselinkname is set to 1, link names are used to identify links in the reports. If this option is used, the link names should be unique.

Simulation Options

frcL3simreroute

(Router models only). If frcL3simreroute is set to 1, all demands will be routed from scratch when performing a failure simulation. This parameter may be desirable in some situations to more accurately simulate load balancing behavior when there are equal cost multiple paths. If frcL3simreroute is set to 0 (default), only failed demands will be routed from scratch when performing a failure simulation.

maxcallsetup

This is a simulation option. It gives the call setup retry count.

simmaxhop

This is a simulation parameter: maximum number hops in a path

siminterval

This option sets the interval time for failure simulation in terms of seconds. The default value for siminterval is 20.

MCsimrptopt

If set to 1, the simulation reports are modified to report multicast demand failure by number and bandwidth of the trees instead of the branches. For example, if more than one branch fails, it is counted as one demand failure and the bandwidth failed is added to the total failed demand bandwidth once instead of multiple times for each branch.

Sizing Options

These options are used during PVC sizing operations (for ATM networks), or LSP tunnel sizing operations (for MPLS-enabled networks).

minSizingBW

The minimum value for any new tunnel/PVC bandwidth. If the calculated bandwidth is less than this value, then this value is used as the new bandwidth.

maxSizingBW

The maximum value for any new tunnel/PVC bandwidth. If the calculated bandwidth is greater than this value, then this value is used as the new bandwidth.

incSizingBW

The increment by which the bandwidth will be increased. Basically, the calculated bandwidth will be rounded up to the nearest multiple of this value.

sizing_growthmultiplier

This value is multiplied by the flow bandwidth to calculate the new bandwidth. For example, 1.00 will generate a new bandwidth that is 100% of the flow bandwidth, and 1.5 will be 150% of the flow bandwidth.

sizing_growthconstant

A constant offset to add in the calculation of the new bandwidth.

sizing_resizeopt

0 = only increase; 1 = fit to traffic. The “Only Increase” option is for sizing only overbooked PVCs/tunnels. When this option is set, a new bandwidth will only be calculated if the flow bandwidth is greater than or equal to the current planned bandwidth. When the “Fit to Traffic” option is set, a new bandwidth will always be calculated.

Hardware-Related Parameters

hwvendor

The parameter hwvendor is used to indicate the hardware device vendor. For users with several hardware models, this parameter should be set accordingly. The default link bandwidth, link overhead and per circuit bandwidth overhead calculations will vary for different vendors.

Valid hardware types include:

GENERIC PACKET, GENERIC CIRCUIT, NET IDNX, NET FRX, NET-STM, DACS, TIMEPLEX LINK2/100, NEWBRIDGE, TDAX, CISCO ROUTER, JUNIPER, ROUTER, BAYNETWORKS ROUTER, BAYNETWORKS BNX, CASCADE FRAME RELAY, CASCADE ATM, STRATACOM, GENERIC ATM, CISCO LIGHTSTREAM1010, OPAT, FORERUNNER ASX, ALCATEL, FOUNDRY, RIVERSTONE, MPLS-TP

mplsenable

If set to 1, all links are MPLS enabled

noBGP

BGP Routing information will be ignored.

phyhoplimit

The physical hop limit (phyhoplimit) is used to indicate the hardware limit for the maximum number of links allowed in a path. The value of maxhop cannot exceed the value of phyhoplimit.

reversepri

Every end-to-end circuit demand requirement must have a priority assigned to it. There are two different ranking methods:

- higher number means higher priority (reversepri = 0)
- higher number means lower priority (reversepri = 1)

In the first case, a circuit demand with priority 5 has a higher priority than a circuit demand with priority 1. In the second case, a circuit demand with priority 5 has a lower priority than a circuit demand with priority 1.

If reversepri is set to 0, than the first interpretation is used. If reversepri is set to 1, the second interpretation is used.

reversepri is automatically set by the program according to hwvendor. It should normally not be set in the parameter file except for experimental purposes. If reversepri is set in the parameter file, the default for the hwvendor is ignored.

tunnelovhd

Tunnel Overhead

Link Bandwidth and Overhead Parameters

The parameters listed under this category are automatically set by the program according to the hwvendor parameter. These parameters do not need to be modified unless the user is emulating other hardware or trunk types.

The capacity and link overhead of trunks types can be modified by setting the corresponding parameters in the parameter file. Supported trunk types include:

```
#T3bw=     40704000  # system default
#T3lkovhd=        0  # system default
#E3bw=     30528000  # system default
#E3lkovhd=        0  # system default
#T1bw=      1544000  # system default
#T1lkovhd= 44000  # system default
#E1bw=      2048000  # system default
#E1lkovhd=        0  # system default
#T2bw=      6143760  # system default
#T2lkovhd=        0  # system default
#OC3bw=    149760000  # system default
#OC3lkovhd=        0  # system default
#OC12bw=    599040000  # system default
#OC12lkovhd=        0  # system default
#OC48bw=   2377728000  # system default
#OC48lkovhd=        0  # system default
```

Informational Note: Some trunk types and values displayed above will not be applicable to all hardware.

The link overhead is the general bandwidth overhead reserved by the hardware. Available bandwidth to applications is the difference between the actual bandwidth and the link overhead. For instance, if the bandwidth of a T1 (T1bw) is 1.544 Mb, and 44 Kb of every T1 is reserved for link overhead (lkovhd), then the usable bandwidth of each T1 is 1.5 Mb.

Per circuit overhead is another factor that needs to be calculated during the bandwidth allocation of circuits. Circuit overhead calculation is hardware dependent.

VLAN Parameters

keepl2sptree

Setting this value to 1 will keep the spanning tree information parsed from the file. Setting this value to 0 will cause the program to be in a “smart” mode. For example, for isolated sections of a spanning tree without a root node, a root node will be selected.

addroute2treename

When setting this value to 1, the spanning tree name in the VLAN view will be followed by the suffix @rootname to indicate the root node of the tree. If one tree is shown as multiple components in the VLAN windows spanning tree view, this is an indication of missing links.

ECMP Parameters

ECMP flag can be set on a demand which will split the demand into smaller aggregate demands. By default the original demand is split into 6 equally sized aggregate demands, or into X equally sized aggregate demands on condition that the minimum aggregate demand bandwidth is greater than or equal to 1M. Example 1, if the original demand is 120M, it will be split into 6 20M aggregate demands. Example 2, if the original demand is 4M, it will be split into 4 1M aggregate demands. The max number of split demands can be defined by the “maxECMPcnt” parameter in the dparam file (default is 6). The minimum aggregate demand bandwidth can be defined by the “minECMPflowbw” parameter in the dparam file (default is 1M).

maxECMPcnt

The max number of split demands. This should not be used with ECMPcntByBW.

minECMPflowbw

The minimum aggregate demand bandwidth. This should not be used with ECMPcntByBW.

ECMPcntByBW

This parameter defines the number of ECMP flows created for an ECMP demand based on demand bandwidth. The format is minbw1:count1|minbw2:count2|...

Example: ECMPcntBW=300M:72|100M:32 means any ECMP demand with bandwidth >= 300M is split into 72 flows, demands with bandwidth >= 100M is split into 32 flows, and demands < 100M is kept as one flow. Default ECMP behavior is to create 6 flows for every ECMP demand, or X flows greater than or equal to 1M, if this parameter is not used.

Miscellaneous Parameters

advFilterThreshold

Originally, the advanced filter for demands and tunnels was performed from the Java client side. For networks with tens of thousands of demands or tunnels, this could be slow in performance because the necessary information had to be downloaded first from the server to the client. Now, the advanced filter feature is available on the server side for a limited subset of demand and tunnel properties.

The advFilterThreshold parameter sets the number of demands and tunnels that must be in the network before server side advanced filter is enabled as opposed to client side advanced filter. For example, if advFilterThreshold is set to 500, then server side advanced filter will only be enabled once there are over 500 demands.

batch

Some of the interactive messages given from bbdsgn may be hard to predict. This makes running a batch process more difficult. Some of this nondeterminism can be avoided if you set the parameter, batch, to 1. Otherwise, the default value of batch is 0.

For example, you may have a number of unplaced demands and bbdsgn may offer to show them, say, twenty at a time. Suppose you want to see them, but without having to be prompted continually for the next twenty demands. If you know how many times you are prompted, you can run a batch process with the given number of responses. But in this case, it is hard to predict. When you set batch to 1, the interactive mode gets turned off for this particular question.

cos2lspmap

Set cos2lspmap parameter to map demands of certain cos classes to LSPs of particular names based on wild card matching. For example, cos2lspmap=MRT:MRT7 means take a demand that belongs to cos class MRT and try to route it on a lsp where the lsp name is MRT7. Another example, cos2lspmap=MC:MC* means take a demand that belongs to cos class MC and try to route it on any lsp where the lsp name begins with MC. If there are multiple lsp’s that matches, then the demand will be load balanced among the lsp’s.

lambdabw

This parameter defines the bandwidth for one lambda unit.

linklatencyvalue

This parameter is used in the online mode to determine whether to use the average, minimum, or max, when reading in collected link latency data from File > Read > linklatency. Possible values are AVG, MIN, or MAX.

matchtunnelname

If set to 1, each demand will be preferentially placed on a tunnel with the same name as the demand name (if any). Note that demands may need to be rerouted from scratch to see this effect (e.g., via "Reroute all Demands" upon opening the network from the Java interface or by selecting **Design > Demands > Route Paths > Route from Scratch**.

Chapter 4

Node Files

Domain File

Description

This file is used to define domains for TDM hardware and OSPF areas for router and Lucent models. Both require an entry in the license file.

Syntax and Example

Each domain entry in a domain file should be on its own line, and have three fields separated by spaces/tabs: domain_ID/area_ID, name, and color. The following example:

```
#domain_ID    domain_name  color
1             REDNET       MAGENTA
V2            BLUENET      BLUE
TRANSIT=V2
```

defines two domains. The color assigned to domain 1, REDNET, is MAGENTA. The color assigned to domain V2, BLUENET, is BLUE. When the nodes are displayed in graphics mode, they are drawn using their domain color. The same three fields apply for area definition.

ID

An alphanumeric string; please limit to 10 characters. This field is used as a unique identifier for a domain or an area. For domain definition, the format Dxxx, where xxx is a number, is reserved to signify a domain with domain number of xxx.

name

An alphanumeric string; please limit to 10 characters

color

RED, MAGENTA, YELLOW, GREEN, BLUE, CYAN, WHITE, BLACK

In general, it is not recommended that RED or GREEN be used for domain colors since the default color for links between gateways is GREEN, and the color of disabled nodes is RED.

TRANSIT=*domain_name*

In the sample listing above, notice the line TRANSIT=V2. This indicates that domain V2 (Domain_Name: BLUENET) is a transit domain. A transit domain is a concept used in net.com's Super WAN routing algorithm, and is required for modeling. It allows multiple domains to be interconnected by routing paths through a core, or transit, domain. Note that a user demand may only span 3 domains, including the transit domain.

This also applies for OSPF areas, where the transit area is usually “AREA0”, which is the backbone area.

Other Usage Info

The transit domain concept can also be applied to hardware devices that follow the Open Shortest Path First (OSPF) routing algorithm. In modeling OSPF “areas”, the transit domain can be used to represent the "backbone". For each OSPF autonomous system, there must be a backbone. Analogous to the net.com implementation, the backbone may be used as a bridge when demands are placed between several areas. A more detailed explanation of the OSPF algorithm can be found in the user manual for the respective hardware device.

Graphcoord File

Description

The graphcoord file contains user-defined coordinates for backbone nodes in the graphics window. This file can be generated by bbdsgn. If defined in the specification file, the contents of graphcoord are set as the default window definition when graphics mode is entered. Graphics coordinates may be changed interactively by moving locations around within bbdsgn.

Usage

This file can be generated by the JAVA Graphical User Interface.

Example

An example graphcoord file is listed below:

```
window  1228  158   2114   1515
#node   npa   nxx   graph_v  graph_h
N001    212   406   4919     1447
N002    212   406   4933     1570
N003    212   406   5154     1394
N004    212   406   5174     1593
N006    212   406   4905     1518
N007    212   406   5218     1501
N008    212   406   5046     1109
N009    212   406   5145     1164
END
```

The first line defines the window size. Only locations and line segments within the window coordinates are displayed.

Group File

Description

Topology groups file: Useful for visual grouping; unlike the site file, this file does not influence diversity design, simulation, path placement, or pricing.

Syntax

```
#   Group_name    Members
       GROUPA     N1, N2, N3
```

Usage

Grouping is a topology feature used to group nodes together. If you save your specification file with groups, the next time you open it up, nodes in a group will be grouped together under one group symbol.

Muxloc File

Syntax

The muxloc file contains the node ID and name of each node in the network. The general format for the muxloc file (United States and Canada locations):

```
#NodeID    name         npa    nxx  [MISC] [COLOCATION]
N01         NYC(5WTC)    212    392
```

For international locations, the following format applies:

```
#NodeID    name   npa    nxx    country_code   latitude   longitude [MISC]
N33        LEED   999    999     UK            534959N    0013459W
```

V and H coordinates can also be used in place of latitude and longitude coordinates:

```
#node_ID   name    npa    nxx    country_code    v  h  [MISC]
```

The following V/H format generated by the WANPricer program is also accepted:

```
#node_ID name      npa nxx lata wcv   wch   rcv   rch   telcoID state  [MISC]
SBTN     SBARRNGTN 708 551  358 05993 03540 05953 03505 5070    IL
```

For the UK, the following UK Ordnance Survey Grid’s Eastings/Northings format is also accepted if saveUKENcoord is set to 1 in the dparam file.

```
#node_ID   name     999   999    UK    UKE     UKN
COV        Coventry 999   999    UK    UKE=436 UKN=283
```

NodeID

The node_id field indicates the node ID and may contain any combination of alphanumeric characters. The field size should be limited to 27 characters. There are certain specifications with special meaning. Please refer to the usage note for more details.

Name

The node name is simply a second means of labeling your node. Typically, the node_id is kept fixed for tracking purposes, but the name can be changed back and forth for other purposes. The field size should be limited to 27 characters. Spaces are not permitted.

npa

For US/Canada. The 3-digit area code. The npa nxx are used by the program to determine latitude and longitude coordinates.

nxx

For US/Canada, the three digits of the phone number following the area code

country_code

Two-letter country code, e.g., US for United States, UK for United Kingdom. For the US, this field can be replaced by the LATA (local access and transport area) number.

Latitude

In certain circumstances, latitude and longitude are used to calculate airline distance for pricing purposes. Latitude can be specified as either

- A floating number between -90 and 90, or
- A string of the format ddmmssN or ddmmssS, where N is for North and S is for South, and “dd”, “mm”, and “ss” are substituted respectively by two-digit degrees, minutes, and seconds.

Longitude

Longitude can be specified as either of the following:

- A floating number between -180 and 180, or
- A string of the format dddmmssE or dddmmssW, where E is for East and W is for West, and “ddd”, “mm”, and “ss” are substituted respectively by three-digit degrees, 2-digit minutes, and 2-digit seconds.

UK Easting, Northing (UK Ordnance Survey Grid)

The UK easting and northing fields are for the UK Ordnance Survey grid and uses a point of origin near Scilly Isles. The field should be specified as UKE=nnn UKW=nnn, where “nnn” is a 3-digit number. For these fields to be saved, saveUKENcoord should be set to 1 in the dparam file. (Note that more digits will be accepted for the easting and northing fields, but for each of these, only the first three digits will be saved.)

Miscellaneous

The MISC field is optional and includes the following keywords and expressions. If more than one are used, use spaces/tabs to separate them. No particular order is required. Some of these are specific to certain optional features which may require licenses.

SINGLE_END

The SINGLE_END field specifies that a node is single ended, that is, it can only be connected to one other node in the network. During a design, bbdsgn connects a single ended node with the closest node in the network if it is isolated.

FAIL=0

The fail field marks a node so that it is not brought down during the diversity design or failure simulation processes, used to specify that a node is reliable

CLASS=*classname*

The CLASS field classifies the node in a user-defined class name. This field is used in combination with the custrate file for pricing purposes. The custrate feature requires a license.

DOMAIN=[D*domain_number*|*DomainID*]

The DOMAIN field is used for the Domain Feature for net.com hardware or OSPF area feature for router and Lucent models. Specifies that a node is in a user-defined domain or area. The node may be directly connected to any nodes of the same domain.

GATEWAY

The GATEWAY field is used for the Domain Feature for net.com hardware or for OSPF area borders (for router or Lucent models). Specifies that a node is a gateway node. The GATEWAY designation allows the program to connect the node to another gateway node in order to route demands going to or through other domains.

GATEWAY=*DomainID1*,*DomainID2*,…

The GATEWAY field is used for the Domain Feature for net.com hardware or OSPF area feature for router and Lucent models. Gateways may be restricted by specifying only those domains or areas which the gateway may connect. This information comes from the Accessible Area List field in the Node window.

AREA=*OspfAreaID*

The AREA field is used for the OSPF Area Feature for certain hardware types, including CISCO, JUNIPER, RIVERSTONE, FOUNDRY, and Lucent (CASCADE). This feature requires a license. Often, OspfAreaID is listed as an IP address.A node may be assigned to multiple areas using the format shown below:AREA=areax,areay,areaz.   The first area in the specification, areax, is selected as the primary area.

Owner=*OwnerID*

The Owner field is used for the Owner Feature. Facilitates identification of node ownership. If a demands owner is not defined in the demand file, it may also be set according to the end nodes' ownership. (See separate manual on the Owner Feature for details.)

CORE

The CORE field is used for a two-level hierarchical design in which links of one trunk type are bought for the core nodes and links of another trunk type are bought for the non-core nodes.

In this two-level design, it is assumed that paths can go from regular nodes to core nodes and back to regular nodes. Once a path goes from a core node to a regular node, it cannot go back to a core node. To turn off this constraint, refer to forceCoreRoute on page 13.

ACCESS

The ACCESS field is used to specify the hardware type. If ACCESS is specified as the hardware type, it will be treated as a node that does not participate in routing. Demands cannot transit through the node, only originate and terminate. If the access node is connected to a few nodes, then during design, the system will only buy links to those nodes if more links are needed. If that node is not connected to any nodes, then during design, the system will only buy links to the geographically closest node. To buy links between the access node and more than one node during a design, a link should be manually added in advance. The behavior is similar for diversity design except that for an isolated node, the system will buy links to the two geographically closest nodes (subject to specified site or node diversity constraints).

SINGLE_END

A single ended node can only connect to one other node.

FEEDER

A feeder node is a node that can only connect to one other node and only through one link. It is a special case of the single ended node.

REGULAR

Nodes that do not fit in the category core, single end, or feeder. This is the default hierarchy level if neither CORE, SINGLE_END, or FEEDER are specified.

Co-location with IXC POP/LEC WC

The pricing formulas used to calculate the private line cost between locations of different types such as POP (Point of Presence), Bridging Wire Center, and LEC (local exchange carrier) T1 MUX locations, and customer sites are usually different. To support this difference in private line pricing to different types of locations, the following notations are introduced:

Notation	Description
ATTPOP	AT&T Point Of Presence
MCIPOP	MCI Point Of Presence
USSPOP	Sprint Point Of Presence
WTGPOP	WorldCom Point Of Presence
LECT1MUX	LEC T1HUB Wire Center
LECBWC	LEC Bridging Wire Center

The above notations may be placed at the end of a node entry. For example, to indicate that N01 is co-located with an AT&T POP, any of the following three formats may be used:

```
N01 NYC(5WTC) 212 392 132 05004 01405 05054 01425 5130 NY ATTPOP
N01 NYC(5WTC) 212 392 ATTPOP
N01 NYC(5WTC) 212 392 US 404241N 0740042W ATTPOP
```

Note that if a location is indicated as LECT1MUX, bbdsgn assumes this location is used to direct low-speed circuits into a T1. As such, the channel termination charge at a LECT1MUX site is set to 0 for low-speed circuits.

Usage

Usage Note for NodeID

There are several NodeID formats with special interpretation. The formats below were added to accommodate the domain feature. In the table, nnn stands for a number, ccc for the card number and ddd for the domain number. “D”, “N”, “C” are capital letters. If a Node ID is described using the Domain-Node-Card format, e.g., D10N02C03, the program will automatically extract the domain number, node number, and card number from the ID. Two IDs are treated as identical if they have the same domain, node, and card numbers; or if they have the same exact character string. For these formats, leading zeros are ignored so nodes specified as N5, N05, or N005 are all treated as the same node.

Special NodeID Format	Naming Convention
Node Number	nnn or Nnnn
Node-Card	NnnnCccc
Domain-Node	DdddNnnn
Domain-Node-Card	DdddNnnnCccc
Character String not in any of the above formats	xxxxx

Another format with special meaning is the node-card-slot-port format. This is a node ID followed by a period followed by S*s*C*c*P*p* where *s*, *c*, and *p* are integers.

Note for Latitude and Longitude

Latitude and longitude may be useful for pricing for U.S. T3 Local Loop Pricing and International Private Line Pricing. For Certain local exchange carriers (LEC), the distance between the customer's location and its serving wire center is used to calculate T3 channel termination costs. Without the latitude and longitude coordinates, the program calculates the T3 channel termination costs using the assumption that the location is one mile from its serving wire center.

Example

Following is an example of a muxloc file:

```
#nodeid      name       npa  nxx  [MISC]
N01         NYC(5WTC)  212  392
D02N9        SBARRNGTN  708  551
N11         BRIDGEWTR  201   722
DAL_TX       DALLAS_A   214  426
NJ05         WANDL      908  580
N10         London     999  999   UK  51.500 -0.167
```

A sample usage of FAIL=0 is shown below:

```
N10 wandl 732 868 DOMAIN=1 FAIL=0
```

The following entry indicates that N10 may only be connected to one other node in the network:

```
N10 wandl 732 868 SINGLE_END
```

Nodeparam File

Description

Node hardware type file: The nodeparam file specifies node types and constraints.

Syntax

```
#nodeID/name hwtype  [MISC]
Y36          MUX
```

NodeID/name

Used to indicate a node from the muxloc file.

Hardware_type:

E.g., ETHERNET, CISCO, JUNIPER, BPX, MUX, IGX, NGS, DMSU, M160, M40, TDAX, ACCESS, etc. Default values (for example, MUX and/or NODE) are used if the nodeparam file is not specified in the specification file. MUX, which represents a generic multiplexer device, may be used for any vendor.

MUX (or NODE)

Default hardware type which can be used for any vendor.

MISC (optional):

BLOCKTRUNK

When this keyword is set, the design function will not add any more trunks to this node.

If a node's "blocktrunk" flag is set, that node is not considered during design. The program tries to route demands through other nodes. If it cannot find any way to buy a link to route a demand, that demand is marked as "fail to buy link to route" and ignored during the design process. The design will continue with other demands.

DELAY=*[number]*ms

Total delay at the node (default=1ms)

IPADDR=*IPAddress*

Substitute IPAddress with the IP address. Used for routers and Lucent switches

MaxTransitCall=100000

Specifies the maximum number of demands/calls that can pass through this node, which does not include any demands originating or terminating at this node.

OSPFREFBW=*bandwidth*

(Routers, IP/MPLS module) OSPF reference bandwidth. Used to calculate the default interface metric.

SHORTCUT=YES

Enables forwarding equivalence class. This field is a part of the tunnel feature and is used for router hardware. The default value used for Cisco routers is YES.

SHORTCUT=NO

Disables forwarding equivalence class. This field is a part of the tunnel feature and is used for router hardware. The default value used for Juniper routers is NO.

SOURCE=*path*

(Routers, IP/MPLS module) Specifies the location of a configuration file corresponding to the node.

SRVCPROF=*service_profile*

This references a service profile in the service profile file associated with the node. The service profile indicates what services are handled by the node.

Protocol

(Routers, IP/MPLS module) For example, OSPF,ISIS,BGP,IGRP,EIGRP

net.com

The following parameters are used by net.com devices. They control per node path placement constraints.

maxhop=n

maximum number of links allowed in paths originating from this node.

maxsathop=n

maximum number of satellite links allowed in paths originating from this node.

preempthop=n

A circuit demand can preempt other low priority paths only if it fails to allocate a path with less than x + n hops, where x is the minimum number of hops in the optimal path.

Example

```
#  Format Example:
#     N001  MUX
#nodeID/name   hardware_type   [MISC]
Y36               MUX        IPADDR=192.168.0.7 SHORTCUT=YES OSPF BGP
AS16631         ASNODE      SHORTCUT=NO
N1           IDNX70       maxsathop=1
```

Site File

Description

Site definition file, useful for diversity design, simulation, path placement, pricing, and so on.

Syntax

```
sitename=node1=node2=...=nodeN
```

Sitename

Choose site names that differ from node names to avoid potential confusion in the input data. The sitename should be a maximum of 31 alphanumeric characters.

Node

A node can be indicated by ID or name. These IDs and names can be found in the muxloc file.

Usage

The site file is used to define sites as logical groupings of nodes. The site definitions are used for the purpose of failure simulation, diversity design, and diverse path placement. Site information can also be used to facilitate pricing specifications, as in the usercost file.

By default, if a node is not included in a user-defined site definition, it is treated as being in a site of its own for these purposes.

Continuing Long Lines with a Slash

```
mysite01=N08
mysite02=N46=N86 = N71 = N72 \
= N73 = N74
```

If you need multiple lines to define a site, use a back slash character (\) to continue the entry from one line to the next.

Example

```
mysite01=N08
mysite02=N46=N86 = N71 = N72 \
= N73 = N74
sboundbrook=N21=N45
mysite04=BRIDGEWTR
```

VPN File (IP/MPLS only)

Description

VPN definition file: Specifies the routers that belong to Virtual Private Networks (VPN) that exists in the network. This feature requires a VPN license.

Informational Note: For IP/MPLS module users only.

Syntax

Layer 3

```
#Type,RouterName,VRFName,RD,Route-Target-Export,Route-Target-Import,VPNID,,protocols,
```

Layer 2 Kompella

```
#Type,RouterName,VRFName,RD,Route-Target-Export,Route-Target-Import,VPNID,,protocols,encapsulation,
siteName,site-ID
```

Layer 2 Martini

```
#2M,nodeA,nodeZ,VCID,circuitA,circuitZ,Encapsulation,vpnName,bw
```

Type

The type of this VPN. Valid entries are: “3” for Layer 3 VPN, “2M” for Layer 2 Martini VPN, and “2K” for Layer 2 Kompella VPN

RouterName

Name of the router that belongs to this VPN.

VRFName

Name of the VPN routing/forwarding instance (VRF) of this router. This field is case-sensitive.

RD

Route-distinguisher.

Route-Target-Export / Route-Target-Import

The route-target specifies a target VPN extended community.

VPNID

Unique identifier of the VPN.

Protocols

The running protocols on the VRF entry.

Encapsulation

Examples of encapsulation types include atm-aal5, atm-cell, atm-cell-port-mode, atm-cell-vc-mode, atm-cell-vp-mode, cisco-hdlc, ethernet, ethernet-vlan, frame, interworking, ppp, aal5, aal0, ppp, hdlc, eth, vlan, fr

Example

```
#Type,RouterName,VRFName,RD,Route-Target-Export,Route-Target-Import,VPNID,,proto
cols,
3,RT_BOSTON_01,Eau,400:1,400:1,10000:1 400:1 ,VPN4,,bgp|connected|static,

#Type,RouterName,VPNName,Near-end-circuitID,EncapsulationID,VCID,Far-end-Router
2M,PE1,111,GigabitEthernet1/1:100,dot1Q,111,4.4.4.4,
2M,PE1,222,GigabitEthernet1/3:100,dot1Q,222,7.7.7.7,
2M,PE2,111,GigabitEthernet1/2:100,dot1Q,111,2.2.2.2,
2M,PE3,222,GigabitEthernet1/2:100,dot1Q,222,2.2.2.2,
```

Bgpnode File (IP/MPLS Only)

Description

BGP speakers definition file: Specifies the BGP speakers that exist in the network. This feature requires a BGP license.

Informational Note: For IP/MPLS module users only.

Syntax

```
#NodeName ASno ConfederationID clusterID misc
```

NodeName

Name of the node (BGP speaker).

ASno

Number identifier of the autonomous system (AS) of the BGP speaker.

ConfederationID

The confederation identifier if it is applicable.

clusterID

The cluster ID if it is applicable.

misc

Miscellaneous information on the BGP speaker such as whether the neighbor is an RR client or not.

Usage

This is a text file that specifies the BGP speakers that exist in the network. Users need to comment out the specification of the bgpobj file in the specification file if they plan to edit BGP attributes manually. When loading the network, the rtserver (or bbdsgn) program reads the bgpobj file, if it is specified, ignoring the bgpnode and bgplink files. However, if the bgpobj file is not specified or it is commented out, rtserver will read the bgpnode and bgplink files instead. When saving the network, all three files: bgpobj, bgpnode and bgplink will be saved.

Example

```
#NodeName ASno ConfederationID clusterID misc
N3        222   0               0          RR
```

Chapter 5

Link Files

Bblink File

Description

The bblink file is a IP/MPLSView file describing the location, quantity, vendor, and attributes of the backbone links.

Syntax

```
[LinkName] Node Node Vendor # BwType Misc
```

LinkName

There is an optional link name field that can precede the From_Node field. A name may be assigned to help identify a particular link. Link names are used in pattern matching as well as in various reports, such as LINKCOST and LINKUTIL. The user may specify whether links should be represented by using their names or their endpoints. Link names may consist up to 39 consecutive alphanumeric characters (no spaces). Dashes (“-”) in the link name are not allowed, and will be converted to an underscore (“_”) when loaded onto the server. Please refer to uselinkname on page 22 on how to turn on the link name in reports.

It is recommended that link names be unique in a given network.

Node

These two node fields specify the nodeID of the link’s endpoints. The nodeIDs must be found in the muxloc and nodeparam files.

Additionally, card/slot information can be indicated using the following format: Node.C*x*P*y* where *x* is the card number and *y* is the slot number. For example, the following is a link connecting card 1, slot 2 of node 1 to card 1, slot 3 of node 2.

```
TRANS_21    N1.C1P2   N2.C1P3      DEF    1  OC192
```

Vendor

The following vendors are supported: ATT, USS, MCI, WTG, LEC, NET, and DEF, where ATT, USS, MCI, and WTG are used to indicate private lines belonging to the IXC vendors AT&T, US Sprint, MCI, and WorldCom, respectively.

- LEC - Local Exchange Carrier
- NET - In-house fiber links that have no associated cost
- DEF - Default vendor

International Vendor Names

Vendors in countries other than the United States may be specified in the bblink file using the same format. As there are numerous possible vendors for all the countries supported by IP/MPLSView, they are not listed in this section. The user should consult the respective manual for the corresponding country/tariff database licensed. Inter-country vendors may be defined using the format XXYY where XX and YY are the vendors for the two countries (ie. MCBT represent MCI and British Telecom).

# (Count)

This integer specifies the number of single lines that exist between the nodepair in the “Node” fields. When the count (#) is greater than one (1), the “LinkName” field must be empty since the “LinkName” field must be unique for each link.

BwType (Bandwidth Type Field)

The following is a partial list of bandwidth types supported by the bbdsgn program: T3, E3, T1, T2, OC3, OC12, OC48, FT56K, FT64K, etc., F-xxxK (xxxK means xxx Kilobits), SATRK (Misc field TX, RX), HSSIxM (xM means x Megabits), AIM_xT1, AIM_xE1, ETxM (xM means x Megabits), ET1G (1G means one Gigabits).

Misc (Miscellaneous Field)

The miscellaneous and media type field contains media and link overhead, distance, cost, delay specifications for links. If more than one specification qualifier is needed for a link in the media type field, these specifications should be separated by commas (,) without any spaces. The specifications can also be separated by spaces.

Field	Possible values	Default	Other Usage Info
Media Types	TERRESTRIAL, SATELLITE, FIBER, ENCRYPT	TERRESTRIAL	Partial string from the beginning accepted (e.g., T for TERRESTRIAL)
Link Overhead*	OVHD=, OVHDA2Z=, OVHDZ2A=	Default unit is in bits.	Specify K, G, M for kilo, giga, and mega.
Link Delay*	DELAY=, DELAYA2Z=, DELAYZ2A=	Default unit is seconds	Specify “ms” for milliseconds. Lower case or Upper case are both accepted,
Link Distance*	DIST=, DISTA2Z=, DISTZ2A=		Note that when a trunk is multiple-defined and a distance is given, the distance applies to all the trunks defined together. This metric is used with the Adm_Weight routing method.
Link Cost	COST=*n[currency_unit]*		e.g., DL for dollars, BP for British pounds
Link Reliability	FAIL=0		

Field	Possible values	Default	Other Usage Info
Link Bandwidth	BW=x, BWA2Z=x, BWZ2A=x		Specifies usable bandwidth if different from the default value.
Link Administrative Status	Install, Install->Delete, Live, Live->Delete, Order,Order->Delete, Planned,Planned->Delete,		The administrative statuses given on the Location tab are to be distinguished from the operational status on the Link Properties tab. While doing design, links marked with no status will be removed if not needed. Links marked with live, planned, order, install will not be removed and are marked as the corresponding status with the "->Deleted" appended at the end.

*Starred fields can have symmetric or asymmetric values.

Link Delay

Routing can be specified based on minimizing the propagation delay associated with physical distance. The default delay for a trunk is based on the propagation delay. Here, Propagation Delay is determined using the assumption that each 100 miles equals 1 ms of delay.

FAIL=0

As mentioned in the analogous section for nodes on FAIL=0 on page 30 of this document, bbdsgn by default fails each node and link during the diversity design phase, but the user has the option of specifying certain nodes and links as permanent or always available. To define a link as always available even during diversity design and failure simulation, the following string needs to be added to the end of the line entry in the bblink file: FAIL=0

ATM-Specific Keywords

These are ATM-specific keywords and values in the miscellaneous field:

Field	Description
CELL	For cell trunk - ATM protocol used
PCT=*traffic_type: linkpct [,traffic_type: linkpct]*	This is the format for specifying maximum percentage of bandwidth for link partitions. *linkpct* should be a floating number from 0 to 1.
MINPCT=*traffic_type: linkpct [,traffic_type: linkpct]*	This is the format for specifying minimum percentage of bandwidth for link partitions. *linkpct* should be a floating number from 0 to 1.
OVF=*Partition_name: x [,Partition_name:x]*	*x* is the overbooking factor (=1/K). Examples of partition names are CBR, VBR, RT, NRT, ABR. Include no spaces within this expression. Include a space preceding and following the expression.
PVC=*n*	*n* is the maximum limit on the number of PVCs supported on each trunk. The limit varies by trunk size and switch type.

Router-Specific Keywords

These are IP/MPLS-specific keywords and values in the miscellaneous field:

Field	Description
Protocol	Indicates that the link is enabled for this protocol. Options include RIP, IGRP, EIGRP, OSPF, ISIS, LDP, TDP, SRP. Multiple protocols can be entered for a link.

MBW=*bw*
MBWA2Z=*bw*
MBWZ2A=*bw*
	Sets the bandwidth used for metric calculation for OSPF or (E)IGRP. Note that this is the routing parameter configured for the interface, not to be confused with the physical bandwidth, which is represented by BW, BWA2Z, and BWZ2A.

EIGRP=*delay*,
EIGRPA2Z=*delay,*
EIGRPZ2A=*delay*
IGRP=*delay*,
IGRPA2Z=*delay,*
IGRPZ2A=*delay*
	Sets the delay value used for metric calculation for EIGRP or IGRP. Note that this is the routing parameter configured for the interface and should be distinguished by the propagation delay, which is represented by DELAY, DELAYA2Z, and DELAYZ2A. The units can be specified as ”us” for microseconds. E.g., IGRP=10000us

ISIS1==*cost*,
ISIS1A2Z=*cost*,
ISIS1Z2A=*cost*
ISIS2=*cost*,
ISIS2A2Z=*cost*,
ISIS2Z2A=*cost*
	Sets the explicit metric for ISIS level1 or ISIS level2

OSPF=*cost*,
OSPFA2Z=*cost,*
OSPFZ2A=cost
	Sets an explicit OSPF link cost for metric calculation.

RI=<name>	Indicates the associated OSPF routing instance name
MPLS	MPLS TE-enabled. A link must be enabled for MPLS traffic engineering in order for traffic engineering tunnels to be routed over it. For tunnel feature.

TDIST=*number*
TDISTA2Z=*number*
TDISTZ2A=*number*
	The link’s metric as seen by tunnels being routed over the link if no IGP metric is specified.

ATTR=*hexadecimal*	Link attributes. For tunnel feature. E.g., ATTR=ffffffff

GLBPOOL=*bw**
GLBPOOLA2Z=*bw**
GLBPOOLZ2A=*bw**
	Cisco-only. Tunnels cannot route over a link unless there is available bandwidth in the global pool.

SUBPOOL=*bw**
SUPOOLA2Z=*bw**
SUBPOOLZ2A=*bw**
	Cisco-only. “Guaranteed bandwidth” tunnels cannot route over a link unless there is available bandwidth in the subpool.

FRR_A=*backuptunnel*
FRR_Z=*backuptunnel*
	Specifies Fast Reroute backup tunnel protecting the link. (Tunnel license is required.)

POLICY=*policyname*	Specifies CoS policy name. (CoS license is required.)
ASLINK	Specifies a link to an Autonomous System.

Field	Description

C1=interface_name1
C2=interface_name2
	This is to indicate the interfaces of the two end points where interface_name1 and interface_name2 are names of the interfaces of node A and node Z, respectively.
	
	There should be no space between the keyword, the equal sign, and the name. Also the names should not include space. For example, the following are incorrect specifications of interface name 1:
	C1 = Serial2/0/0
	C1=Serial 2/0/0
	The following is the correct specification:
	C1=Serial2/0/0

IP1=ip_address1[/subnet_mask1]
IP2=ip_address2[/subnet_mask2]
	ip_address1 and ip_address2 are IP addresses of the interface_name1 and interface_name2, respectively. Here, *subnet_mask1* and *subnet_mask2* are the subnet masks and are optional. E.g., `IP1=192.10.20.218/30`

POLICY1=policy_name1
POLICY2=policy_name2
	This is to specify the CoS policies applied to the above interfaces. policy_name1 and policy_name2 are names of the policies applied to the interface_name1 and interface_name2, respectively.

For other hardware-specific or feature-specific file format information, please see the relevant documentation.

Usage

Each entry defining a link should consist of at least five fields: From_Node, To_Node, Vendor, Quantity, and Type. These five fields may be separated by either spaces or tabs.

Example

```
FLR2    N10    N8    NET    1     T1    FAIL=0
      N2     N6    ATT   3  T3
      N1    N2  ATT  1   T1   Fiber DIST=100 $2000 OVHDA2Z=180K
      N1    N2  ATT  1   T1   Fiber,DIST=100,$2000,OVHDA2Z=180K
      N1    N2  ATT      1    T1   Delay=4
Link345 Paris2 London4 DEF 1 C1=Serial2/0/0 C2=Serial5/0/1 IP1=192.10.20.218/30
IP2=192.10.20.217/30 POLICY1= polA1 POLICY2=polZ3
```

Bgplink File (IP/MPLS Only)

Description

The BGP link file is a text file that contains information on the BGP neighbors. The BGP feature requires a BGP license.

For IP/MPLS module users only.

Informational Note: Due to the complexity, peer group and policy are not defined in these two files now.

Syntax

```
#linkID nodeA nodeZ Z_AS MED weight local_preference multi_hop RRClient
```

linkID

Unique identifier of the link that is used as a BGP link.

nodeA

The name of the BGP speaker.

nodeZ

The name of the BGP neighbor of nodeA.

Z_AS

The AS number of the neighbor, nodeZ.

MED

The Multi-Exit Discriminator attribute.

weight

The weight attribute.

local_preference

The local preference attribute.

multi_hop

The optional TTL (Time to Live) number from the IOS command:

neighbor {ip-address | peer-group-name} ebgp-multihop [ttl]

RRClient

The indicator to indicate whether the neighbor is an RR client or not.

Usage

The bgplink file is used to define BGP neighbor information. Users need to include the specification of the bgplink file in the spec file to apply it to the network model. When loading the network, the rtserver (or bbdsgn) program reads the bgplink file, if it is specified.

Example

```
#linkID nodeA nodeZ Z_AS MED weight local_preference multi_hop RRClient
NBR1   N1       N2       111    0        0          0         0
```

Delay File

Description

The delay file is a file used to update information in bblink, including the link latency, bandwidth, and metric. It can be specified during config extraction as a getipconf option via the -delay <delayFile> option.

Syntax

```
#!NodeA,Interface,LatencyA2Z,BW
LDN2600,Ethernet0/1,50,100m
ATL,fe-0/1/3.0,50,100m
```

Usage

The format of the link latency file is flexible. The customizable column headers should be specified in a comma-separated list following a "#!". The column headers on this line must be one of the following reserved keywords in order to be recognized.

- **NodeA, NodeZ, Interface, InterfaceZ**
- **LatencyA2Z, LatencyZ2A**: Latency from NodeA to NodeZ (ms) and vice versa, from NodeZ to Node A (ms). For microseconds, use decimals.
- **RoundTripLatency**: This number will be divided by two to get the latency
- **BW-K**: The bandwidth in Kbps
- **BW**: The bandwidth in bits
- **ISIS2Metric**: The ISIS level 2 metric

Note that the data for one link could also be represented in one line instead of two. For example, the above link latency file entry for the link between LDN2600 and ATL could be shortened to one line by including the LatencyZ2A column, as shown below:

```
#!NodeA,Interface,LatencyA2Z,LatencyZ2A,BW
LDN2600,Ethernet0/1,50,50,100m
```

The RoundTripLatency can be specified as an alternative to the Latency in one direction.

```
#!NodeA,Interface,RoundTripLatency,BW
LDN2600,Ethernet0/1,100,100m
```

For backwards compatibility, the following fixed format is also supported:

```
#RouterA,Type,RouterZ,Interface,Interface IP,Bandwidth(K),Metric,LatencyZ2A
conf1,,,Ethernet0,10.0.0.1,,,10
```

For the fixed format, the only attributes required are RouterA, Interface, and Latency. Note that the direction of Latency here is from NodeZ to NodeA.

Example

```
#!NodeA,Interface,RoundTripLatency,BW
LDN2600,Ethernet0/1,100,100m
```

Facility File

Description

The facility feature is an optional add-on feature that requires a license. The facility file lists all defined facility names, as well as links and/or nodes associated with that facility.

Syntax

```
#Hostname:FacilityName FacilityType NodeOrLinkName1 NodeOrLinkName2 etc...
```

Hostname:Facility Name

Hostname is from the device and Facility Name is a user-defined name.

Facility Type

Required to use keyword "fate-sharing".

Node Name

Nodes may be specified either by their node ID or node name. If both are used in the same facility, then that node will be duplicated. Nodes which are not in the mux file and links not in the bblink file are ignored.

Link Name

The name of a link in the bblink file.

Usage

In this file, the first field defines the facility name. The subsequent fields specify the node IDs or link names associated with that facility, delimited by tabs, spaces, or commas. All elements associated with a facility should be specified on the same line. Whenever more than one line is needed to specify the elements, a backslash, '\', must be used to indicate that the element list is continued on the next line.

The facility feature does not check the validity of the nodes and/or links listed in the facility file. Duplicate links and/or nodes will also be duplicated in the facility.

Example

```
#Hostname:FacilityName FacilityType NodeOrLinkName1 NodeOrLinkName2
CHI:FAC10 fate-sharing LINK1 LINK7
CHI:FAC21 fate-sharing NODE3 LINK8
NYC:FAC30 fate-sharing LINK8 LINK10 NODE20
```

For more information regarding the facility file, please refer to the Facility Feature document.

Linkdataupdate (Delay/Bandwidth/Metrics File)

Description

The linkdataupdate file is a file used to update information in bblink, including the link latency, bandwidth, trunk type, and metric. It can be read in through File > Read menu (via the file type **linkdataupdate** under the **Device Specific Files** section. Click the **Browse** button to indicate the location of the file to use for updating the links.

Alternatively, in a console window, type /u/wandl/bin/bbdsgn *specfilepath.* Select from the Main menu: **5. Modify Configuration > 4. Link Configuration > u. Update Link Properties from a File**. Select **2. Input File Name** and enter in the location of the linkdataupdate file (absolute or relative path is acceptable here). Select **5. Update link configuration** to perform the actual update based on the linkdataupdate file. To save the changes, use the Main menu’s **2. Save Files** option.

Syntax

```
#!NodeA,Interface,DelayAZ,BW
ATL,fe-0/1/3.0,50,100m
```

Usage

The format of the link latency file is flexible. The customizable column headers should be specified in a comma-separated list following a “#!”. The column headers on this line must be one of the following reserved keywords in order to be recognized.

The first line should specify the columns using a comma separated list of the above keywords, including a column for the node and the interface or IP address at the minimum. The subsequent lines should specify the Node/Interface or Node/IP pair and the other relevant columns to update.

- **NodeA, NodeZ, Node,**
- **InterfaceA, InterfaceZ, Interface, IPaddrZ, IPaddr**
- DelayAZ, DelayZA, Delay: Latency from NodeA to NodeZ (ms). For microseconds, use decimals.
- **RoundTripDelay**: This number will be divided by two to get the latency
- **LinkName**
- BWType: E.g.,ET100M, ET1G
- **BW**: The bandwidth in bits
- **OSPFMetric, ISIS2Metric, ISIS1Metric**: The OSPF, ISIS level1, and ISIS level 2 metric

Example

```
#!NodeA,Interface,DelayAZ,BW
LDN2600,Ethernet0/1,50,100m
ATL,fe-0/1/3.0,50,100m
```

Linktemplate File

Description

The facility feature is an optional add-on feature that requires a license. The facilit file lists all defined facility names, as well as links and/or nodes associated with that facility.

Syntax

```
#TemplateName  AutoName  LinkPrefix  NameIndex  LinkName  NA
NZ  Vendor  BwType  Misc
Default        No        -           0          -       -   -   DEF     T1
```

Template Name

Name for the template.

AutoName

Yes or No. Specifies whether the links should be given names automatically with a link prefix and index number.

Link Prefix, NameIndex

If AutoName is yes, then LinkPrefix is the prefix of the link name and NameIndex is the starting number for the automatic naming. For example, if Link Prefix is MyLink and NameIndex is 5, the links added will be named MyLink5, MyLink6, MyLink7, etc.

LinkName

Specifies a particular link name. If auto naming will be used, this field can be left blank using the ‘-’ symbol.

NA NZ Vendor BwType Misc

These fields are the same as those of the bblink file.

Usage

This file is used to define link templates that can be used to make adding new links easier.

Example

```
#TemplateName  AutoName  LinkPrefix  NameIndex  LinkName  NA NZ  Vendor  BwType  Misc
MyTemplate     Yes       MyLink      1          -         - N01  DEF     STM1 OSPF,BW=1M
```

Policymap File (IP/MPLS Only)

Description

The policy map feature is a feature that allows you to specify Class of Service (CoS) policies. The policymap file is used to list the mapping of classes to policies and routers. Each line is for one policy of a router. One router can have several policy-maps. Each line in the policy map file contains information about the policy name, router name, defined classes and class policies (such as bandwidth and queue length). The priority class is always listed before the other classes.

The CoS feature requires a CoS license and is for IP/MPLS module users only.

Syntax

```
# Type|GlobalParameters|Router_name|Policy_name|Priority_class_name, bandwidth, - |{class_name,
bandwidth, queue_length, dhcp_bitmap, dhcp_bitmap_cont|}
```

Type

Type of queuing algorithm. Valid types are CBWFQ, MDRR, MDRR strict, MDRR alternate, and HWRR. Note that HWRR is represented by either ERX_NODE or ERX_QUEUE in the policymap file.

GlobalParameters

This field is currently only used with HWRR policies to define hierarchy levels.

Router_name

Name of the router on which the policy is specified.

Policy_name

Name of a policy.

Priority_class_name

Name of the Priority class.

bandwidth

Bandwidth for the priority class. The units is in Kbps. Alternatively, the user may specify a percentage of the total BW in this field. This number must be followed by a ‘%’ symbol. If the Type field is specified as MDRR, MDRR strict, or MDRR alternate, then users should input the MDRR relative Weight value for the priority class into this field.

class_name,bandwidth,queue_length,dhcp_bitmap

This field defines the policy for each class. It is repeatable for up to six classes not including the priority class. For the specified policies, provide the CoS class name and the corresponding guaranteed bandwidth (in either Kbps or a percentage of the total BW) and queue packet limit. A ‘-’ for the bandwidth or queue length indicates default values. If the user specifies a percentage value in the bandwidth field, it must be followed by a ‘%’ symbol (e.g. 30%). The dhcp_bitmap field is used to match class names with DS-TE LSP class types.

The priority class is for Low Latency Queuing or Priority Queuing. Packets belong to this class have higher priority than other classes. There is no queue limit for this class. That is why there is the dash “-“ in the third subfield.

Example

```
CBWFQ|
|BRU|smallBW|voice,64,-|first_class_data,300,32,0,0|business_data,200,16,0,0

MDRR strict|
|BB|BB_smallBW|voice,64,-|first_class_data,300,-,0,0|business_data,230,-,0,0|

ERX_QUEUE|LEVEL=1|ERX_NODE|POS_PROFILE|-,-,-|best-effort,16,0,0,0|ef-class,80,0,
0,0|af31-class,10,0,0,0|
```

Routeinst File (IP/MPLS only)

Description

The route instance feature is a feature that allows you to define OSPF routing instances or process IDs, which can be referenced from the bblink file. See the Router Guide chapter, “Routing Instances” for more details.

Syntax

```
#RIname pid1 pid2 pid3 ...[Color=String]
```

RIname

The name of the OSPF routing instance.

pid

The process ID number. The length of the process ID should be less than or equal to 8.

Color=<String>

An optional specification of the color to be associated with the process ID. Possible values include BLUE, RED, GREEN, WHITE, and CYAN

Example

```
#RIname pid1 pid2 pid3 ...[Color=String]
BLUE_70  70 BLUE community=1234:5678|2856:6123|12641:6123 color=BLUE
RED_60   60 RED community=1234:5677|2856:6124|12641:6124 color=RED
WHITE_12641 12641 NONE color=GREEN
NONE_WHITE_12641 color=BLACK
```

Chapter 6

Demand and Traffic Files

Demand and Newdemand Files

Description

There are two demand files that may be used to read in circuit requirements. Both files have the exact same format. The first file, which is read in as demand file, contains the circuit demands and path specifications needed for the initial network. The second file, if needed, is read in as newdemand file. It is useful when adding additional demands without having to modify the original demand file.

Syntax

The format of a demand file includes the following information: DemandID, From_Node, To_Node, Speed, Type field, Priority, and Path Specification. The Type field may contain multiple subfields separated by commas. Information pertaining to circuit status, bandwidth overhead, and path routing are specified in this field.

```
DemandID   FromNode ToNode Bandwidth Type_field    Priority,Preempt Path
I000123A   N01      N02    256000    R2,SLIVE,DDG1 12,10            N01-N05-N02
```

DemandID

The DemandID, which may have a maximum of 39 characters, is used to identify the demand.

FromNode and ToNode

From and To node information can be defined by node ID or name. Note that the format for identifying the From node may be different from that used to identify the To node.

Bandwidth

This field defines the bandwidth required by the demand. It should be specified as numbers without any commas. Overhead is automatically calculated by the program and should not be included.

Type_field

The parameters in the type field should be comma-separated. Note that not all parameters in the type field are applicable to all hardware devices. Please consult the companion manual specific to your hardware device to determine those parameters. Parameters that are not applicable to a particular hardware device will be ignored by the program.

Priority,Preempt

The priority field of the demand specification consists of two numbers separated by a comma (,), or a forward-slash (/). The first number defines the call priority of the demand, and the second number the preempt priority of the demand. The preempt priority should be at the same or lower priority as the call priority of the demand. It is assumed that this demand can bump any of the demands with call priority lower than the preempt priority.

Path

The path is not a required field. It consists of a sequence of node IDs or names separated by @,@@,&, -, --, =, ==, or [ ]. Different delimiters are used to mark the distance relationship between nodes. This field does not impact routing. It is for informational purposes only. The user should use the PS and PR flags in the type field to specify a preferred or required path.

Delimiter	Description
@	two nodes in different domains, but same LATA /country
&	two nodes in same site
-	two nodes in same LATA/country
=	two nodes in same LATA/country, second link between nodes is used
@@	two nodes in different domains and different LATA/country
--	two nodes in different LATA/country
==	two nodes in different LATA/country, second link between nodes is used
[ ]	path between two nodes that is routed through a tunnel in a IP/MPLS network. The following example indicates the flow FlowXYZ is routed through a tunnel called Tunnel99 between N1 and N10: FlowXYZ N1 N10 10M R,A2Z 02,02 N1[(T=Tunnel99)--N2--N3--N4--]N10

The example path N01-N05-N02 on page 20 indicates that both circuits in this specification are routed from N01 to N05 to N02. All three nodes are in the same LATA. If the path field is specified in the circuit definition, the demand is placed in the backbone according to the path specification.

Type_field in More Depth

Valid Type_field values (separated by commas)

Data, Voice, and Voice Demand

R*cnt*

Regular data circuit, followed by the quantity (count). Default value of cnt is 1. The R and V (see next type definition) types can be used to specify the number of circuits in a demand. If neither R nor V is specified, then type R is assumed.

V*cnt*

Voice circuit, followed by the quantity (count). Default value of cnt is 1. The R (see previous type definition) and V types can be used to specify the number of circuits in a demand. If neither R nor V is specified, then type R is assumed.

VD*cnt*

Demand Voice circuit, followed by the quantity (count). Default value of cnt is 1. A demand voice circuit is a voice circuit allocated on demand. They are disconnected when the voice call is hung up. The V and R type circuits defined previously are permanently nailed circuits.

Symmetric and Asymmetric Duplex Demands and Simplex Demands

A2Z or Z2A

A circuit is called one-way if traffic through it moves in only one direction. If traffic moves from the From_Node to the To_Node only, the circuit should be marked as A2Z. If traffic moves from the To_Node to the From_Node, the string Z2A should be used. A circuit that is not explicitly marked as A2Z or Z2A is assumed to be a full-duplex circuit.

Usage Note

To specify a duplex symmetric demand, leave out A2Z and Z2A in the demand entry.

To specify a duplex asymmetric demand, create two adjacent demand entries both with all the same fields except for the fields that are asymmetric. This includes preserving demandID, NodeA, and NodeZ fields. Then specify the direction for one as A2Z and the direction for the other as Z2A.

To specify a simplex demand, use one of the directions A2Z or Z2A. There should be no adjacent entries with the same demandID, nodeA, nodeZ that would cause it to be categorized by the program as a duplex asymmetric demand.

Other

QoSname

(ATM users only) e.g., CBR,RT,NRT,ABR,UBR,UBR+

RBAL=n

(Lucent users only) Balance Rerouting: n=0 ignore, n=1 enable. Switch will constantly check for better route and if one is found, it will switch.

MC

Multicast. The multicast feature requires a license. For PVCs to use the multicast feature, the DemandID, FromNode, direction (A2Z, Z2A), bw (bandwidth), QoSname, peak, mean, and burst, should be the same

H*hopcount*

This is the maximum hop count limit for the circuit (e.g. H8 indicates a eight maximum hop count)

MAXCOST=*x*

If a Max Cost value is set for a demand, it will only be placed if there exists a path with an aggregate administrative cost equal to or less than the Max Cost value.

MAXDELAY=*x*

When a Max Delay value is set for a demand, it will only be placed if there exists a path with a delay value equal to or less than the Max Delay value. This delay is based on the sum of the node and link delays. Link delay is based on user's specification or physical distance. Note that unless the user specifies the delay time unit, the default time is milliseconds. The user may also use "ms" to specify milliseconds or "s" to specify seconds.

Informational Note: Older versions have seconds as the default unit.

This parameter is for design purposes. It ensures the maxdelay constraint is kept for the shortest path in the case where there are no tunnels in the network. This constraint is not currently being applied to the case where the actual path goes through a tunnel or if the actual path is a user-configured path.

LDP

When LDP flag is set, the demand can only be routed over LDP enabled links or over the TE with LDP enabled. For example, if multiple tunnel choices are available, only the tunnel enabled with LDP protocol will be selected. This demand can also be routed over tunnels with NOAA flag (no auto-announce).

ECMP

When ECMP flag is set, by default the original demand is split into 6 equally sized aggregate demands, or into X equally sized aggregate demands on condition that the minimum aggregate demand bandwidth is greater than or equal to 1M. Example 1, if the original demand is 120M, it will be split into 6 20M aggregate demands. Example 2, if the original demand is 4M, it will be split into 4 1M aggregate demands. The max number of split demands can be defined by the "maxECMPcnt" parameter in the dparam file (default is 6). The minimum aggregate demand bandwidth can be defined by the "minECMPflowbw" parameter in the dparam file (default is 1M).

M*media_pref*

Media preference is marked using the letter M. Three types of media preference are supported: terrestrial (T)/microwave (M)/satellite (S), fiber (F), and encryption (E). For each media type, the user can specify whether it is required (R), Preferred (P), or preferred not (N), or avoid (A).

Usage is illustrated in the following examples:

- MTP: Terrestrial Preferred
- MTR: Terrestrial Required
- MTN: Terrestrial Not Preferred
- MTREP: Terrestrial Required, Encryption Preferred

O*owner*

The character O may be used to mark the owner of a circuit. The purpose of the owner feature is to facilitate the identification of demand ownership. By defining an owner and associating certain demands with that owner, the task of bandwidth reconciliation is simplified. Service providers that carry the traffic of several companies can use the owner feature to quickly determine the distribution of traffic in the network.

PBK*n*(*backup_path*)

For backup routes. In the demand type field, PBK*n*(N1-N2-...-N3) stands for a backup path with Opt value *n.* For example, put PBK3(A-B-C) for backup path from A to B to C with Opt value of 3. The *Opt* field represents the priority of the path.

Sample demand entry with preferred path A-B and backup path A-C-B:

```
#demandID  NodeA  NodeZ  BW  Type                            Pri,Pre
RAB        A      B     10M  R,Z2A,PNNI,PS(A-B),PBK1(A-C-B),CBR  02,02
```

PR(*path)* or PS*(path)*

The type PR ("path required") indicates a required path. PS ("path select") indicates a preferred path. When a circuit demand is assigned a required path (PR), it will only route on that specified path. If the path is not available, the circuit demand will not be placed. On the other hand, when a circuit is assigned a preferred path (PS), it will attempt to route on that specified path first. If the path is not available, the circuit demand will attempt to place on an alternate path.

The path can consist of node, link or IP. However if links are used, it is required to include the starting nodeA connecting to the link, otherwise the routing will be incomplete. For example, PS(AS1-NODE1-LINK1**NODE8-LINK8-AS2). The path starts from AS1, connects to NODE1, continues through LINK1, ** indicates the path goes through a loose route, connects to NODE8, continues through LINK8, and ends at AS2.

SRVC=*servicetype*

The service type in the servicetype file associated with the demand.

S*status*

Status is the status of the circuit. The acceptable values for status include LIVE, PLANNED, INSTALL, DELETE, NEW.

U*pct*

Specifies the utilization percentage. Default value is 1.0 (100%). Only available if hardware supports it.

Call by Call Simulation

Informational Note: This feature requires a call by call simulation license.

DATE(begindate-enddate),TOD(begintime-endtime)

Specifies multiple events. E.g., DATE(11/5/97-11/11/97),TOD(12:05-13:10) In this case, during the simulation run, a demand with this time value will be active only during the period between November 5, 1997 and November 11, 1997 each day between 12:05 PM and 1:10 PM.

TIME(begindate@begintime-enddate@endtime)

Single event. E.g., TIME(10/5/97@6:00-10/11/97@23:59) During the simulation run, a demand with this time value will be active only between October 5, 1997 at 6:00 AM and October 11, 1997 at 11:59 PM. If the TOD begintime and endtime are equal to each other, the call-by-call simulation will assume a 24-hour period. Likewise, if the TOD endtime is less than the begintime, the call-by-call simulation will assume a simulation past midnight (i.e 12PM - 1AM).

Discrete Event Simulation

Informational Note: This feature requires a discrete event simulation license.

Trafficname

trafficname points to traffic descriptions in the traffic file. It is used during the packet-by-packet simulation (e.g., TTYPE1). This feature is optional and requires a license.

Diversity Groups

D*divgp*

To ensure that a group of demands is routed in site- and edge-disjoint paths, they can be defined as one diversity group by assigning them a diversity group name. A diversity group name should be a maximum of 7 alphanumeric characters. Here is an example:

Dmd1 N1 N2 R,Dgroup1

Dmd2 N1 N2 R,Dgroup1

Both demands are in the same diversity group, group1.

Informational Note: The name, from, and to, fields can be the same, but do not have to be.

The diversity group name can be marked in the type field of the circuits in the diversity group by adding D*divgp* to the type field, where *divgp* is the name of the diversity group. (Be sure not to name it such that D*divgp* is a reserved word. For example, Date or Div will be treated as reserved words and not diversity groups.)

Demands marked with the same diversity group name may be partitioned into pairs.In the reports, these subgroups will have an extension. For example, if there are six demands in mygroup, the subgroups will be named mygroup.1, mygroup.2, and mygroup.3. Demands in the same pair will be routed in site-disjoint paths. As such, any site or link failure can bring down only half of the circuits in a diversity group.

The diversity group level (site or edge diversity) can be specified in the dparam file. Set divgrouplevel to 3 for site diversity and 2 for edge diversity.

The group name “SITEDIV” is reserved. Demands (or tunnels for layer 2) marked to be in this group will be paired with other demands (or tunnels) of this group with the same origination and destination sites.

Secondary Paths and Standby Paths

Secondary

This parameter is only for hardware devices that support the secondary route feature. Indicates that the demand is a secondary demand to be referenced by a primary demand.

A secondary demand is not routed until the primary demand fails. For example:

CKT1001 N1 N2 20M R,SECONDARY,A2Z 05, 05

DIV and STANDBY

Standby demands and tunnels are routed whether or not they are used. If the primary demand fails, this path is used. A standby demand should have the same name, from, and to parameters as the primary demand.

In bbdsgn, you are allowed to have one hot standby per demand. This hot standby can be manually entered or automatically created for you by bbdsgn.

In the demand file, the entry for a demand/tunnel's hot standby should always immediately follow the demand/tunnel entry.

The demand/tunnel entry for which there is to be a standby demand/tunnel, will have the keyword DIV in the type field. The demand/tunnel entry for the hot standby will have the same name, from, and to fields. Instead of the keyword DIV, the STANDBY keyword will appear in the type field.

For example,

```
Circ1 N1 N2 R,DIV
Circ1 N1 N2 R,STANDBY
```

For automatic generation of the standby entry, you only need to specify the demands and tunnels for which you want a hot standby, with the DIV keyword. You do not need to manually enter in entries for the hot standby. bbdsgn can then generate the hot standby demands and tunnels for you, upon reading in the demand file. The hot standby for a demand/tunnel will immediately follow the entry for that demand/tunnel for which you wanted a hot standby. It will have the same name, from, and to fields, and the STANDBY keyword instead of the DIV keyword.

In the case that you have something like:

```
Circ1 N1 N2 R3, DIV
```

the demand will be split up into three demands, each with a standby immediately following.

Note that if the hot standby is already manually entered in or generated by bbdsgn, bbdsgn will not create another one.

Router Demand Parameters

cos_class

The CoS class for the demand.

ATM Demand Parameters

RM=*rmtype*

(ATM users only) Defines the routing method used for this demand definition. *rmtype* values may vary from hardware to hardware. Examples:

- ADM: Shortest path is based off of distances defined in the bblink file (DIST). If distance is not defined, the default distance is set to 100
- OSPF: Shortest path is based off of distances defined in the bblink file (DIST). If distance is not defined, the path cost formula of OSPF is used
- DELAY: Shortest path is based off of DELAY instead of DIST, defined in the bblink file

BP*speed*

Port speed, unit = bits per second

BR*speed*

Peak rate, unit = bits per second

BM*speed*

- Mean Rate, unit = bits per second, or
- Minimum Rate, unit = bits per second (ABR,UBR+)

BB*size*

Bburst size or duration

DIVSEC

DIVSEC is a flag indicating that a secondary path should have been created for this PVC. The program will add a secondary path entry if missing.

DIVSTBY

DIVSTBY is a flag indicating that a hot standby path should be created for this PVC. The program will add a standby path entry if missing.

PVC

Indicates nailed-down demands for Marconi. During a failure, bandwidth for nailed-down demands will not be freed even when the demand is down. To indicate partially-nailed down the PVC parameter can be used in combination with the PR (path required) specification to indicate which portions of the path are nailed down and which are not. Use ** to indicate a loose route. For example, “PVC,PR(A-B-C**D-E-F)” indicates that the paths A-B-C and D-E-F are nailed down but the path from C to D can be rerouted if failure occurs.

RR

Re-routable. This is a convenient way to indicate that if a PVC is unable to route according to its other specified routes, then the originating node will search for a path not following the configured routes. This is equivalent to setting up a secondary route that is Dynamic.

QoS*name*

Quality of service name(CBR,RT,NRT,ABR,UBR,UBR+)

Frame Relay Demand Parameters

BCsize

Size equals the committed burst size in bits (Frame Relay services) or bytes, or average burst size (other services)

BEsize

Size equals the excess burst size in bits (Frame Relay Service) or byte

BFsize

(average) frame size, unit = bytes

net.com (net.com Only)

B*type*

Circuit types that require additional bandwidth overhead are indicated with a B. The following notations are used. Note that this parameter is only for net.com devices.

- BT - Transparent signaling
- BA - Asynchronous timing
- BP - Pass-through timing

Access Design (Access Design Only)

Q*vendor*

Vendor is the default vendor of the circuit. This field is used in evaluating the cost of an offnet circuit demand. Supported vendors include: ATT, USS, MCI, WTG, and LEC. The Q field is used only for circuit pricing in Access Design.

CoSAlias File (IP/MPLS Only)

Description

Informational Note: The CoS feature requires a license.

The CoSAlias file contains mapping information among different config files which have different class definitions. This file needs to be specified in the *specification file* with the entry “CoSAlias=*file_created*”.

The program supports up to 7 classes within each policy map. The first one is reserved for class-default and the second one for the priority queue (or low-latency) class. The other five classes can be put in any order. If the user has different names for the same queue entry (aliases), they should be grouped together in the CoSAlias file.

Syntax

```
#Alias  class1  class2 ...
```

Alias

Name of the general class of service the CoS’s are being mapped to.

Class1, class2 ...

Classes of service that is mapped to the more general categories of service.

Example

```
#Alias  class1  class2 ...
data1  priority-data  core-business-data  class1-data
voice  core-voice  hybrid-voice
data2  business-data  class2  class3
```

Interface File Outbound (IP/MPLS Only)

Description

The outbound interface file contains outbound traffic data for network interfaces broken down in intervals, up to a maximum of 24 periods. This data is used to display the traffic load for links in a network model. For an offline network, this file needs to be specified in the *specification* file with the entry `interfaceLoad_out=interface.`*`runcode`*`.`

Alternatively, in the GUI, you can read in this file by going to **File > Read** and clicking on the **“Egress**” button.

Informational Note: For the live network model in the Management and Monitoring package, this file is named “interface.traffic”

Syntax

The following are some alternative formats:

Standard Format

```
NodeID  InterfaceName  Direction AvgFrameSize  Period1  Period2  Period3 ...
```

For Interface Traffic

```
NodeID  InterfaceName  IPaddr  AvgFrameSize  Period1  Period2  Period3 ...
```

For Cisco Class of Service Interface Traffic

```
NodeID  InterfaceName  IPaddr  Classname  Period1  Period2  Period3 ...
```

For Juniper Class of Service Interface Traffic

```
NodeID  InterfaceName  IPaddr  IP:Classname  Period1  Period2  Period3 ...
```

Multicast Interface Traffic

```
NodeID  InterfaceName  IPaddr  MULTICAST:  Period1  Period2  Period3 ...
```

NodeID

ID of the node that the interface is coming out from.

InterfaceName

The name of the interface, e.g., Ethernet2/0

Direction

Direction of the traffic. In this case, for outgoing traffic from the interface, this field is always “A2Z”.

IPAddr

This is the IP address used to collect data for the node. If the NodeID is unrecognized, this IP address will be used to determine the node.

AvgFrameSize

Average frame size is typically used in the traffic load files for converting payload to actual load, by taking overhead into account. For interface load, however, this conversion is not needed; simply put a ‘-’ or ‘0’ in this field. It is assumed that overhead is already included in the interval definition.

Classname

For Cisco, use the CoS class name. For Juniper, you can either use “IP:classname” or “IP:queuenumber” where the queuenumber is the number of the queue from 0 to 7.

Period1 Period2 etc.

The remaining columns (Period1, Period2, etc.) indicate the traffic load (bps) measured during the corresponding interval. A maximum of 24 intervals may be specified per interface.

Unit = number

This field may be placed before the actual interface traffic data. The default unit value is 1 bit. All the traffic data in the traffic load file is interpreted as kilobits if the value of unit is specified as 1000. The data unit is interpreted as byte if unit = 8 is specified.

Example

```
UNIT = 1
NODE1 FastEthernet0/13 A2Z 0 878 871 872 843 845 856 858 846 847 860 837 822 823 836 810 811 890 1045
945 913 867 825 833 822
NODE1 FastEthernet0/19 A2Z 0 39739 303225 16505 844 10591 10405 712 10442 835 11608 14572 2456 9059 1099
11108 745 11434 47923 360744 143248 131317 338451 678116 26686
NODE2 FastEthernet0/1 A2Z 0 892 879 886 863 864 877 876 866 866 880 854 841 843 855 829 831 889 1030
10124 7201 3891 837 841 829
NODE2 FastEthernet0/2 A2Z 0 893 882 888 864 867 877 885 865 870 881 855 843 845 855 830 832 891 1038 952
916 877 837 842 832
NODE2 FastEthernet0/25 A2Z 0 1359 1382 1444 1370 1397 1319 1550 1437 1429 1422 1549 1434 1455 1414 1879
1471 1516 1431 7786 254511 266246 1380 673500 1337
NODE3 ATM1/0.1 A2Z 0 192320 204960 30263 12893 32227 12693 12240 285327 250747 156934 12701 11802 25981
19664 18697 13602 18763 47406 45705 414819 364185 97485 169042 106109
NODE3 FastEthernet0/0 A2Z 0 321288 833310 161567 12539 14550 12960 11337 65934 1865306 651858 11479
11202 31599 406136 329883 13231 46192 621633 415218 790061 641931 657181 224433 278520
J3 ge-0/0/0 192.168.1.3 IP:best-effort-f 3212 8330 1667 139 150 1260 1137 6534 18656 6858 114 112 319
4066 3283 132 461 6213 4158 7901 6431 6181 2433 2520
C5 Ethernet2/0 192.168.1.4 IP:class-default 3212 8330 1667 139 150 1260 1137 6534 18656 6858 114 112 319
4066 3283 132 461 6213 4158 7901 6431 6181 2433 2520
C8 Ethernet1/0 192.168.1.5 MULTICAST: 3212 8330 1667 139 150 1260 1137 6534 18656 6858 114 112 319 4066
3283 132 461 6213 4158 7901 6431 6181 2433 2520
```

Interface File Inbound (IP/MPLS Only)

Description

The inbound interface file contains inbound traffic data for network interfaces broken down in intervals, up to a maximum of 24 periods. This data is used to display the traffic load for links in a network model.

This file needs to be specified in the *specification* file with the entry `interfaceLoad_in=interfacei.`*runcode*.

Alternatively, in the GUI, you can read in this file by going to **File > Read** and clicking on the **“Ingress**” button.

Syntax

The following are some alternative formats:

Standard Format

```
NodeID  InterfaceName  Direction AvgFrameSize  Period1  Period2  Period3 ...
```

For Interface Traffic

```
NodeID  InterfaceName  IPaddr  AvgFrameSize  Period1  Period2  Period3 ...
```

For Cisco Class of Service Interface Traffic

```
NodeID  InterfaceName  IPaddr  Classname  Period1  Period2  Period3 ...
```

For Juniper Class of Service Interface Traffic

```
NodeID  InterfaceName  IPaddr  IP:Classname  Period1  Period2  Period3 ...
```

Multicast Interface Traffic

```
NodeID  InterfaceName  IPaddr  MULTICAST:  Period1  Period2  Period3 ...
```

NodeID

ID of the node that the interface is going into.

InterfaceID

ID of the interface.

Direction

Direction of the traffic. In this case, for incoming traffic from the interface, this field is always “Z2A”.

AvgFrameSize

Average frame size is typically used in the traffic load files for converting payload to actual load, by taking overhead into account. For interface load, however, this conversion is not needed; simply put a ‘-’ or ‘0’ in this field. It is assumed that overhead is already included in the interval definition.

Period1 Period2 etc.

The remaining columns (Period1, Period2, etc.) indicate the traffic load (bps) measured during the corresponding interval. A maximum of 24 intervals may be specified per interface.

Unit = number

This field may be placed before the actual interface traffic data. The default unit value is 1 bit per second. All the traffic data in the traffic load file is interpreted as kilobits if the value of unit is specified as 1000. The data unit is interpreted as bytes per second if unit = 8 is specified.

Example

```
UNIT = 1
NODE1 FastEthernet0/13 Z2A 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
NODE1 FastEthernet0/19 Z2A 0 19194 14559 19228 637 8562 7964 524 7994 613 8804 11318 1544 7254 659 8598
537 8503 24975 24829 138126 11543 33232 31709 19878
NODE1 FastEthernet0/2 Z2A 0 4125 5809 2225 1737 1424 1709 1403 1422 1162 1366 1829 1535 2311 1748 1146
1776 2036 15366 60686 20801 43106 16672 9080 3660
NODE1 FastEthernet0/23 Z2A 0 56637 323440 16719 7519 11658 9923 7635 63602 1868890 651650 7637 6682
25892 7747 8410 8249 16783 94011 1876140 505526 481805 506694 700195 139386
NODE2 FastEthernet0/1 Z2A 0 4 2 4 3 2 4 5 7 4 6 6 4 5 6 7 4 4 5 1082 2186 1078 3 3 0
NODE2 FastEthernet0/2 Z2A 0 13 12 13 13 14 11 28 6 31 11 11 12 12 11 13 12 12 21 13 13 13 12 13 14
NODE3 ATM1/0.1 Z2A 0 321322 830937 161320 12650 14669 13157 11292 66305 1862819 650596 11612 11593 31619
405768 329641 13363 46195 621218 414593 789547 640612 656325 224008 278348
```

Owner File

Description

Owner names defined in the muxloc and/or demand files are automatically added to the owner list if they are not defined in the owner file.

Syntax

```
#ID  Name  Color
G1  wandl   blue
G5  wandl2  red
```

ID

Please limit the Owner ID to 12 characters or less, at least one being an alpha character.

Color

Valid Color Values: Red, Green, Cyan, Blue, White, Magenta, Yellow

Pathtable File

Description

The pathtable file contains pathnames and the corresponding routes for those paths. This can be referred to from the demand or tunnel file. Instead of specifying a configured route for a demand/tunnel, you can specify a pathname from this file. Static routes can also be indicated in this file.

Once a pathtable is specified in this file, a pathname can be referenced by the source node of that path in the demand or tunnelfile entry’s type field.

In the specification file, this file is specified by the keyword “`route=pathtable.`*`runcode`*”.

Syntax

```
#NodeName Pathname PathSpec
```

NodeName

Name of the node that the path is starting from

Pathname

The name of the path

PathSpec

The path specification. This should consist of the path specification from source node to destination node delimited by dashes (‘-’).

Two asterisks (‘**’) signifies a loose route. This means that the path is flexible between the two nodes. For example, “A**D” means that from A to D, it can go through either B or C, but it must eventually reach D.

Alternatively, if the path uses the Cisco exclude IP address feature, it should be “EXCLUDE” followed by the elements to be excluded.

Example

```
#NodeName Pathname   PathSpec
TL        atltobos  ATL-44.44.44.44-55.55.55.55-BOS
TL        mypath    EXCLUDE-66.66.66.66-77.77.77.77
WASHDC static*10.10.10.11/32 WASHDC**RWDCBOS
```

In the last example, there is a static route configured at the 10.10.10.11 IP address of WASHDC, that goes through tunnel RWDCBOS.

Srvcprofile File

Description

The srvcprofile file defines the service profiles in a network. A service profile is a collection of service types distributed by the weight of each service type. It can be used as a template when creating a new set of demands.

Syntax

```
Profile_Name   Service_Type   Weight
```

Profile_Name

Name of the service profile. This field need not be unique. Each service type assigned to this profile will have the same Profile_Name.

Service_Type

The name of the service type that is a part of this service profile. This must be defined in the srvctype file under Srvctype_Name.

Weight

The weight of this service type in comparison to the other service types in this service profile. It can be given as a percentage or a count. The weights of each service type in the profile will be normalized so that they are distributed accordingly.

Example

```
# profile name     service type     weight
  p1               telnet           50.00
  p1               ftp              50.00
  p2               voice            15.00
  p2               video            60.00
```

Srvctype File

Description

The srvctype file defines the service types in a network. A service type is a category for demands that specifies different characteristics of a demand. It can also be used as a template when creating new demands. The parameters of this file are very similar to that of the demand file.

Syntax

```
SvrcType_Name   MinBW   MaxBW   Bandwidth   Type_field   Priority,Preempt
```

SvrcType_Name

Name of the service type. This must be a unique value.

MinBW

The minimum bandwidth for any demand of this service type.

MaxBW

The maximum bandwidth for any demand of this service type.

Bandwidth

This field defines the bandwidth of a demand that is created of this service type. It should be specified as numbers without any commas. Overhead is automatically calculated by the program and should not be included.

Type_field

The parameters in the type field should be comma-separated. Note that not all parameters in the type field are applicable to all hardware devices. Please consult the companion manual specific to your hardware device to determine those parameters. Parameters that are not applicable to a particular hardware device will be ignored by the program.

Priority,Preempt

The priority field of the demand specification consists of two numbers separated by a comma (,), or a forward-slash (/). The first number defines the call priority of the demand, and the second number the preempt priority of the demand. The preempt priority should be at the same or lower priority as the call priority of the demand. It is assumed that this demand can bump any of the demands with call priority lower than the preempt priority.

Example

```
telnet     256K        1M          256000  R,SRVC=telnet,RT 03,03
ftp        128K       256K         128000  R,A2Z,SRVC=ftp,RT 03,03
```

T_trafficload (Tunnel Traffic Load File) (IP/MPLS Only)

Description

The T_trafficload tunnel traffic load file is used to read in measured tunnel bandwidth utilization based on data collected from IP/MPLS-enabled networks.

Informational Note: The tunnel trafficload features requires a special license.

In the *specification* file, include the line, "*T_trafficload* = "

Syntax

```
[NodeID:]TunnelName Direction  AvgFrameSize Period1 Period2 etc…
```

[NodeID:]TunnelName

If the tunnel name is unique in the network, only the Tunnelname needs to be specified. However, if the tunnel name is not unique in the network, that is, the same tunnel name exists on two different nodes, then it should be preceded by the NodeID followed by a colon. The node ID and tunnel name should correspond to those defined in the *muxloc* and *tunnel* files.

Direction

As tunnels are unidirectional, use 'A2Z' in this field to indicate One-way direction from Origination switch to Destination switch.

AvgFrameSize

For tunnel traffic, traffic load is measured at ingress points. If an average frame size is specified, then transport layer overhead for different transmission types is added to estimate the actual load on the links. AvgFrameSize may be specified using either of the following conventions:

- -: Tunnel traffic load specified will not be adjusted. It is assumed that overhead is already included in the interval definition.
- #_bytes_in_frame: Tunnel traffic load specified will be adjusted based on the transmission type it is being routed over.

Period1 Period2 etc.

The remaining columns (Period 1, Period 2, etc.) indicate the tunnel traffic load (by default, in bps) measured during the corresponding interval. A maximum of 24 intervals may be specified per tunnel. (The intervals do not have to represent hourly intervals.)

Letters such as "K" and "M" can be used, as in "12.3K".

UNIT = number

This line may be placed first in the tunnel traffic load file above the lines of actual traffic data for tunnels. The default unit value is 1 bit. If this value is specified as 1000, then all the traffic data in the traffic load file is interpreted in kilobits. The data unit is interpreted in bytes if unit = 8 is specified.

Informational Note: Be sure to use the exact format “UNIT = n”, with spaces around the equals sign.

Example

```
UNIT = 1
N1:tunnelA A2Z - 100K 93K 26K 50K 51K 38K
N1:tunnelB A2Z - 12800  12800  12800  12800   12800  12800
N2:tunnelB A2Z - 6852   2083   1372    2749   1183   1242
```

Trafficload File

Description

The trafficload file allows the user to measure the bandwidth utilization based on end to end demand data collected from ATM, Frame Relay, or Router networks.

Informational Note: The traffic load features require a special license.

For ATM networks, a demand corresponds to a PVC. For router networks, end to end demands or flows can be defined in order to perform what-if scenario analyses. The trafficload file lists each demand, followed by the period-by-period peak load experienced by the demand during each time interval for which data was collected.

There are two possible syntaxes that can be used for the traffic load file format. The first field in each entry should be identified by either a demand/circuit ID, or else contain more specific node/card/port information. Basically, this identifier must be unique in order for IP/MPLSView to accurately incorporate the traffic data. For example, for an ATM PVC, a node/shelf/card/port identifier can be specified in the first field, if it uniquely defines the PVC in the network. However, because a port may have several channels, it is necessary in some networks to specify the ATM VPI and VCI in order to have a unique identifier.

Syntax

Two possible syntaxes can be used for the traffic load format:

```
DemandID Direction  AvgFrameSize Period1 Period2 etc…
```

or

```
FromNodeCardPort Direction AvgFrameSize Period1 Period2 etc…
ToNodeCardPort Direction AvgFrameSize Period1 Period2 etc…
```

DemandID

The demand ID must correspond exactly to one defined in the demand file. Note that to ensure accuracy of the traffic load information, demand IDs should be unique.

FromNodeCardPort/ToNodeCardPort

If node card and port information is used to define the demand source and destination within the demand file, the second traffic load syntax may be used. This may prove useful, for example, if demand IDs are not defined in the network or do not uniquely identify the demands in the network. The FromNodeCardPort or ToNodeCardPort values indicate the direction for which the traffic load data was collected -- simply the outgoing direction from the node/card/port. Therefore, the Direction value (second field) in the second syntax is actually ignored; it is kept in the format simply for consistency.

Any of the following formats for FromNodeCardPort and ToNodeCardPort are acceptable as long as they: 1) Uniquely identify the demand, and 2) Exactly match the source and destination values defined for the associated demand in the demand file.

```
FromNode.S<num>C<num>P<num>
FromNode.C<num>P<num>
FromNode.S<num>C<num>P<num>/<VPI>.<VCI>
FromNode.C<num>P<num>/<VPI>.<VCI>
```

where S = Shelf, C = Card, P = Port, VPI.VCI (ATM only) = Virtual Path Identifier.Virtual Channel Identifier.

ATM Format and Example

(Both the From and To entries should be specified in the traffic load file for a two-way PVC whose traffic load differs in both directions.)

```
NODEA.C3P1/512.158 - - 25669.9 502.68 1344.47 ..
NODEB.C3P2/510.220 - - 168672.39 131510.67 144067.19 ..
```

where the From node is NodeA, the To node is NodeB, and Dmd100 is defined in the demand file as:

```
Dmd100 NODEA.C3P1/512.158 NODEB.C3P2/510.220 906c R,NRT.. 2,2
```

Direction

Use ‘-’ for Two-way, ‘A2Z’ for One-way from Origination switch to Destination switch, and ‘Z2A’ for One-way from Destination switch to Origination switch.

Note that in router networks, demands, or flows, are one-way. PVCs in ATM networks can be either one-way or two-way, though two-way PVCs are more common. When using the DemandID traffic load syntax, if the traffic load differs on each direction of a two-way PVC, two entries for this PVC need to be specified in the trafficload file in order to capture the differing A2Z and Z2A traffic loads. For example:

```
Dmd200 A2Z   - 6852   2083   1372    2749   1183   1242
Dmd200 Z2A   - 18795  11703  4578    5065   4748    6155
```

Informational Note: For the FromNodeCardPort/ToNodeCardPort syntax, the Direction field (second field) is actually ignored since the NodeCardPort is used to automatically derive the outgoing direction. The field is kept in the format merely for consistency.

AvgFrameSize

Indicates the average frame size of the PVC. It may be specified using either of the following conventions:

- -: Traffic load specified will not be adjusted. It is assumed that overhead is already included in the interval definition.
- #_bytes_in_frame: Traffic load specified will be adjusted based on whether it is being routed over a frame or cell trunk.

Period1 Period2 etc.

The remaining columns (Period 1, Period 2, etc.) indicate the traffic load (bits) measured during the corresponding interval. A maximum of 24 intervals may be specified per demand. (The intervals do not have to represent hourly intervals.)

UNIT = number

This line is optional and may be placed first in the traffic load file above the lines of actual traffic data for PVCs (see example below). The default unit value is 1 bit. If this value is specified as 1000, then all the traffic data in the traffic load file is interpreted as kilobits. The data unit is interpreted as byte if unit = 8 is specified or as cells if unit = 424 (for 53 bytes*8bits/byte).

Informational Note: Be sure to use the exact format “UNIT = n”, with spaces around the equals sign.

FORMAT

This line is optional and typically begins with “FORMAT DIR FRAMESIZE” (see example below). It should be placed above the first line of actual data. The names of the periods (Per1, Per2, etc) in this line can be modified. These names will be reflected in the traffic load charts.

Example

```
UNIT = 1
FORMAT DIR FRAMESIZE Per1 Per2 Per3 Per4 Per5 Per6
F0001 A2Z   87   6852   2083   1372    2749   1183   1242
F0001 Z2A   456  18795  11703  4578    5065   4748    6155
D0016  -     20   9019   3675   3676    3685   3671   3677
V0001  -     20   12800  12800  12800  12800   12800  12800
```

Trafgen_def File

Description

The traffic generation definition file is a file that is used by the Traffic Generation tool that defines specific settings for generating new traffic in a network. This file is saved after the user specifies the setting in the Traffic Generation tool, and can be reloaded later.

Example

```
# IP/MPLSView file: TRAFFIC GENERATION
# Add Type         Add Amount   Src Type   From   Dest Type   To    Apply Type        Apply
  Total_Volume    100M         Node      Any    Node       Any   Current_Traffic  All
```

```
Total_Circuits  50          Node       ATL    Node       CHI   Profile_Pattern  p1
Total_Circuits  35          Node       DEN    Node       SDG   Profile_Pattern  p2
%_Volume        150%        Node       Any    Node       WDC   Current_Traffic  http
%_Circuits      200%        Node       ATL    Node       Any   Current_Traffic  email
```

Syntax

Add Type

Type of traffic to be generated; two main types: volume and circuits, in percentage or total value

Add Amount

Amount of traffic type

Src Type

Type of source for generated traffic: Node, Site, VPN

From

A specific source node, or any node

Dest Type

Type of destination for generated traffic: Node, Site, VPN

To

A specific destination node, or any node

Apply Type

Generated traffic based on current traffic or profile pattern (such as service type or service profile)

Apply

Service type or service profile to apply, or all

Trafgen_out File

Description

The trafgen_out file is a demand file of the forecast traffic that is created by the Traffic Generation tool. It contains comments describing the rules of traffic generation, then lists the demands, which are organized by the traffic entry from which they were generated.

Syntax

The file begins with a commented description of the rules that are used to calculate how demands are generated based on the definition settings. The actual definition as set in the Traffic Generation tool, or trafgen_def file, is commented next. The remainder of the file separates each entry of the traffic definition and lists the demands that were generated, followed by the subtotal in that section. The end of the file then prints the total number of demands that are listed in this file.

Tunnel (Tunnelfile) (IP/MPLS Only)

Description

The Tunnel file defines the Layer 2 traffic engineering tunnels for the network model. In the specification file, it is specified using “tunnelfile=” followed by the name of the tunnel file.

The tunnel feature requires a license. (IP/MPLS users only.)

Example

```
#ID  NA   NZ    BW  Type                                   Pri,HP
TE1  N5   N13  15M R,A2Z,AFFINITY=409a0001,MASK=1a040650   01,00
TE2  N5   N13  10M R,Z2A AFFINITY=409a0001,MASK=1a040650   01,00
TE3  N5   N14  10M R,PS(N5-N8-N12-N14),0BW                 03,00
TE4  N5   N15   5M R,PR(N5-N8-N12-N15),RM=Adm_Weight       04,00
TE5  N5   N16   8M R,RM=DELAY                              02,00
TE6  N8  N20 10M R,Z2A INCANY=00000003,EXCLUDE=00000004   01,00
TE100 N1 N2 120M R,A2Z,PR(N1-N6-N2),SEC=BCK100 02,02
BCK100 N1 N2 80M R,A2Z,SECONDARY 03,03
```

Syntax

ID

Tunnel ID

NA

From Node

NZ

To Node

BW

Bandwidth size of the tunnel.

Type

Comma-separated list of parameters.

Pri, HP

Priority and Holding Priority. The range of values for priority and holding priority is 0 to 20, where 0 is the highest priority. A holding priority of 0 prevents the tunnel from being bumped.

Type Fields

0BW

(Cisco only). Indicates zero backup bandwidth

A2Z or Z2A

Indicates direction of tunnel. A2Z indicates tunnel direction from origination to termination. Z2A indicates tunnel direction from termination to origination.

ABS=*number*

Specifies the absolute tunnel metric.

Affinity=*hexadecimal*, MASK=*hexadecimal*

Format to specify link attribute requirements for Cisco routers. Example: Affinity=00000001,Mask=0000ffff

(Format for Juniper now uses INCALL, INCANY, and EXCLUDE.)

ATTR

Sets tunnel affinity attributes in hexadecimal format.

CCC

(Juniper only.) Specifies Circuit-cross-connect. This means the tunnel is cross-connecting between two interfaces using CCC.

DIVSEC

DIVSEC is a flag indicating that a secondary path should have been created for this LSP tunnel. The program will add a secondary path entry if missing.

DIVSTBY

DIVSTBY is a flag indicating that a hot standby path should be created for this LSP tunnel. The program will add a standby path entry if missing.

EXCLUDE=*hexadecimal*

See INCANY=hexadecimal, INCALL=hexadecimal, EXCLUDE=hexadecimal on page 72

FRR

Indicates that the tunnel subscribes to Fast Reroute (FRR) protection

FRRLK

Indicates that the tunnel is a Fast Reroute Link Backup tunnel for link protection

FRRND

Indicates that the tunnel is a Fast Reroute Node Backup tunnel for node protection

GB

(Cisco only.) Guaranteed Bandwidth-TE. GB Tunnels can only be routed on trunks with available bandwidth in the SubPool.

GRE

Generic Router Encapsulation

H*number*

Specifies the maximum hops.

INCANY=*hexadecimal,* INCALL=*hexadecimal*, EXCLUDE=*hexadecimal*

Indicates admin group requirements for the tunnel. From Juniper statements *include-any*, *include-all,* and *exclude*.

E.g., INCANY=00000003,INCALL=00000000,EXCLUDE=00000004

MASK =*hexadecimal*

See Affinity=hexadecimal, MASK=hexadecimal on page 72

MAXDELAY=*number*

Specifies the maximum delay allowed on trunk during routing (sum of node and link delays). The total delay cannot exceed this number. The default unit is milliseconds. Users can append a “s” to change the unit to seconds or an “ms” to make it explicit that the unit is in milliseconds.

Informational Note: Older versions have seconds as the default unit.

This parameter is for design purposes. It ensures the maxdelay constraint is kept for the shortest path in the case where there are no user configured paths. This constraint is not currently being applied to the case where the actual path is a user-configured path.

NOAA

No Auto-route Announce

NOBD

No Border flag. This is an artificial parameter used for design. When set, routing will not follow OSPF constraints. That is, the whole network will be treated like a flat network.

NOCSPF

Indicates that administrative groups will be ignored by this tunnel.

PATH*n*(*pathname*) or PATH*n*(Dynamic)

Specifies that the primary path *pathname* where *pathname* is a pathname from the pathtable file or specifies that the primary path is dynamic. The number *n* refers to the priority (for Cisco).

PBK*num*(*pathname*), PBK*num*(Dynamic)

Specifies the backup path *pathname* where *pathname* is a pathname from the pathtable file or specifies that the backup path is dynamic. The number *num* indicates the opt-value (priority) for the path. Used for Cisco tunnel backup routes.

PS(*path)* or PR*(path)*

PS(*path*) is for Path Select or preferred route. User-defined route for the tunnel. If path is invalid, tunnel will be routed by the hardware default. Specified paths are enclosed by parenthesis and are represented by node names, node IDs, link names, or link IP addresses, separated by a dash.

PR(*path*) is for Path Required or fixed route. User-defined route that the tunnel must follow, or else not be routed. Specified paths are enclosed by parenthesis and are represented by node names, node IDs, link names, or link IP addresses, separated by a dash.

R

Set tunnel type to data

REL=*number*

Specifies the relative tunnel metric.

RM=*type*

Specified routing method of the tunnel. If RM is not specified, the default routing method is used. Valid RM types include: DELAY, Adm_Weight, Actual_Mileage, and Constant.

RR

Re-routable. This is a convenient way to indicate that if a tunnel is unable to route according to its other specified routes, then the originating node will search for a path not following the configured routes. This is equivalent to setting up a secondary route that is Dynamic.

Standby

(Juniper only.) Standby is used in conjunction with Secondary to indicates that the secondary tunnel is in standby mode. Standby tunnels are routed while the primary tunnel is up.

Secondary

Indicates that the tunnel is a secondary tunnel to be referenced by a primary tunnel. Secondary tunnels are not routed until the primary tunnel fails.

TMLT=*templatefile*

Specifies a tunnel template to be used for LSP configlet generation.

VT

Virtual Trunk (If tunnel is marked as a virtual trunk, it is known to other routers and its admin weight, available bandwidth information will be broadcast to other routers.).

Trafficpattern File

Description

The traffic pattern file allows the user to define several class types based on traffic characteristics. In the specification file, it is specified using “trafficpattern=” followed by the name of the traffic pattern file.

The traffic pattern feature requires a license.

Example

```
#traname        #msg    Duration        Msg size        framsize
#                       second          bits            bytes
PATTERN1        1.0     2.0             160000          1500
PATTERN2        3.0     1.0             2000000         256
PATTERN3        4.0     3.0             500000          1000
PATTERN4        1.0     1.0             1000000         1000
```

Syntax

TRANAME

Traffic pattern name

MSG

Number of messages to be sent within the duration of the traffic pattern

DURATION

Duration of the traffic pattern in seconds

MSG SIZE

Size of each messages in bits

FRAMSIZE

Frame size in bytes

Trafficdata File

Description

The traffic data file allows the user to define each permanent virtual circuit (PVC) by specifying multiple packets and packet sizes. Although this requires the user to have a reasonable knowledge of the traffic, more accurate simulation results can be obtained in this manner. In the specification file, it is specified using “trafficdata=” followed by the name of the traffic data file.

The traffic data feature requires a license.

Example

```
#format = unit  unit_size
#interval = x (number of seconds)
#pvcname  direction  #unit  unit_size  #unit2  unit_size2
format = packet  size
interval = 300
PVC1 A2Z 16,48,30,512,35,256,20,512
PVC1 Z2A 10,48,20,512,30,256
PVC2 A2Z 20,48,50,512
PVC2 Z2A 20,48,50,512
```

Syntax

Note that although the PVC definition entries in the example above are delimited by commas, the user may also use spaces and tabs to separate entries. For each PVC, up to 10 pairs (#unit and unit_size) may be specified in defining the circuit.

FORMAT

Format of the PVC definition. Valid entries for the format line in the traffic data file are shown below:

```
packet  size
byte    size
bit     size
size    packet
size    byte
size    bit
```

where #unit is the number of units (defined in the format line) in the interval, and unit_size is the size in bytes of those units.

Interval

Time interval in which all the packets defined in this file are sent in

PVCNAME

PVC (permanent virtual circuit) name

DIRECTION

Direction of the traffic. Valid entries for the direction field in the pvcname definition line are as follows:

```
- # bidirectional traffic
A2Z # from node A to Z
Z2Z # from node Z to A
```

#UNIT

Number of units (defined in the format line) to be sent within the time interval

UNIT_SIZE

Size of each unit (defined in the format line) in bytes

Usertunneldef File (IP/MPLS Only)

Description

The usertunneldef file defines design parameters for the Layer 2 traffic engineering tunnels for the network model. In the specification file, it is specified using “usertunneldef=” followed by the name of the user tunnel definition file.

The tunnel feature requires a license. (IP/MPLS users only.)

Example

```
#NodeName,TunnelName,DivGroupName,user-defined-typefields
ATL,RATLCHI,555,MAXCOST=111,MAXDELAY=222ms,H3,Pair=444,3DIV,FACDIV
```

Syntax

Each line is delimited by commas and contains the head-end router, tunnel name, and diverse group name, followed by a list of user-defined type fields.

NodeName

Head-end router

Tunnel Name

ID/Name of the Tunnel

DivGroupName

The tunnel’s Diverse Group. By placing several tunnels in the same diverse group, they can be paired off, and designed to be on diverse paths during the tunnel diverse path design.

user-defined-typefields

The remaining parameters have the following format:

```
MAXCOST=<number>
MAXDELAY=<number>ms
H<number>
Pair=<name>
3DIV
FACDIV
```

Chapter 7

Cost Files

The cost files associated with bbdsgn are specifications of user-defined tariff rates and discounts that are used to define and calculate network costs when a network is designed or modified. Each file type is described below:

File Type	Description	Category and Page
custrate	Defines tariff rates for links that cannot be priced out using the tariff database.	Custrate File on page 77
intlratespec	Specifies volume discount for selected international carriers and services.	Intlratespec File on page 78
ratespec	Specifies year, plan, and volume discount percentages for selected IXC vendors and services.	Ratespec File on page 78
usercost	The usercost file is used to define the cost for links according to the end nodes, vendor, and trunk type.	Usercost File on page 79
usercountrycost	Same as the usercost file except the only difference is that the *from* and *to* fields are replaced by two-letter country codes rather than node IDs.	Usercountrycost File on page 79

Order of Precedence

The order of precedence is: bblink>usercost>usercountrycost>custrate or standard tariff database.

Custrate File

Description

User-defined tariff rates. Requires custrate license.

Example

```
DIST= MILE         #DIST= Unit of measure; KM or mile
COUNTRY= IT        #Specifies the Country Rate
INCR_DIST=0.1 MILE #Incremental unit:0.1 MILE (default=1 M)
#service class1 class2 NRC   [tomile fix rate_per_inc_dist]*
OC3      ALL    ALL    1000  9999 270000 600
T3       CA     CB     1000  100 300 30  9999 1000 20
```

Usage

Enabling the Custrate File

To turn on the custrate file specified in the specification file, set custrate=1 in the dparam file.

File Format

The custrate file is used to define tariff rates for links that cannot be priced out using the tariff database. For any link, the rate depends on the service used and the class of the nodes of the link's endpoints. The service field denotes a trunk type like OC3 or T3. The fields class1 and class2 are logical groupings of nodes. These classes are defined by adding a CLASS=*classname*

in the miscellaneous field of the muxloc file, substituting *classname* with the class name.

The pricing method used is the band method. For a given link, the non-recurring charge is NRC and the recurring charge depends on the mileage band it falls under.

In each custrate entry, following the first four fields (service, class1, class2, NRC) are a set of three numbers representing the first mileage band. Subsequent triplets represent subsequent mileage bands. The field `tomile` indicates the end of the mileage band.

The recurring charge for a link is determined using a fixed rate plus a rate per incremental distance, as shown in the following:

```
recurring charge =
fixed_rate+(rate_per_inc_dist/inc_dist)*link_airline_distance.
```

Example: Suppose we have a 200 mile T3 link between a node in class CA and a node in class CB. Using the custrate file specified above, the link would fall within the second mileage band. The non-recurring charge would be 1000. The recurring charge would be 1000+(20/0.1)*200=41000.

Intlratespec File

The intlratespec file is used to specify volume discount for selected international carriers and services. This file can be created by the "International Tariff Table Modification" in the "Cost/Tariff Modification" menu in the text version of bbdsgn.

```
#Country    Vendor   Discount
FR          ALL       0.9
UK          BTEL     0.85
```

Ratespec File

(For U.S.) The ratespec file is used to specify year, plan, and volume discount percentages for selected IXC vendors and services. This file can be created from the U.S. IXC Rate Plan/Volume Discount option from the Cost/Tariff Modification Menu in ASCII mode.

```
Vendor ServiceType  Plan    Discount Percentage
ATT    ALL          3-year  1.000
MCI    T1           1-year  0.900
```

In the above example, a 3-year plan is used for all AT&T private lines. MCI T1 pricing is based on the 1-year plan with a 10% discount on the mileage rates. The default rate tables in the ratedir directory are used for vendors and services not defined in this file.

For the vendor and services specified in the ratespec file, bbdsgn creates the corresponding rate tables using the database files in /u/wandl/db/rates/default. Note that if there are changes other than for the year-plan rate, or volume discount, the WANPricer program can be used for a customized tariff table.

Usercost File

The usercost file is used to define the cost for links according to the end nodes (or sites), the vendor, and the trunk type. The format of the usercost file is as following:

```
#from    to     vendor     type    [cost/mo.]
N01      N02      ATT      T1      1000.00
N01      N02      ATT      FT56K   300.00
N01      N02      WTG      FT56K   250.00
N03      N04      USS      T1
site2    site3    USS      T1      2000.00
N05      site3    USS      T1      2000.00
N05      site3    ATT      T1      2200.00
```

The from and to fields can specify a node ID, node name, or site name. The user should ensure that site names are different from node names in order to prevent pricing inconsistencies.

If a site name is entered, the listed price and default vendor assignment applies to all locations in that site. The default intra-site cost is 0. In the above example, the cost for any ATT T1 from N05 to any node in site3 is $2200 per month. The cost for any USS T1 from N05 to any node in site3 is $2000 per month.

If more than one vendor’s cost is given for the same node pair and trunk type, then the least cost vendor is treated as the default vendor. In the example above, the default vendor for a FT56K link between N01 and N02 is WTG. If ATT is selected as the vendor for a FT56K between N01 and N02, the cost of that link is calculated as $300.

During a design, created links will be assigned the vendor with the least cost. In this example, the default T1 vendor between N03 and N04 is US Sprint. During a design run, if IP/MPLSView decides to buy a T1 between N03 and N04, it will assign USS as the vendor of that T1. The program will then calculate the cost of this T1 using the default tariff files since there is no cost specified for this entry. Similarly, if buying a link between N01 and N02, the default T1 vendor that will be used is AT&T with a cost of $1000 per month.

Usercountrycost File

```
#CountryCodeA   CountryCodeZ  vendor  type    [cost/mo.]
FR     GE            DEF     FT64K   1000.00
```

The only difference from the countrycost file is that the from and to fields are replaced by two-letter country codes rather than node IDs.

Chapter 8

Control Files

The control files associated with bbdsgn are specifications of user-defined parameters which influence routing and design and specify defaults. Each file type is described below:

File Type	Description	Category and Page
fixlink	Specifies the links that should not be deleted during backbone link design.	Fixing Links on page 83
linkdist	Specifies the default administrative weights for links between backbone nodes.	Setting the Administrative Weight on page 81
nodeweight	Specifies the node weight and the maximum link bandwidth capacity from that node.	Node Constraints on page 83
rsvbwfile	Defines reserved bandwidth for specific node pairs.	Reserved Bandwidth on page 84
admincost	Defines the default administrative weights for links according to specified characteristics of the link.	Setting the Administrative Weight on page 81

Setting the Administrative Weight

Administrative weight (also known as distance or administrative cost) can be specified in the bblink, linkdist, dparam and admincost files.

Turning on the Use of Administrative Weights for Routing

To be turned on, set hopdist=Adm_Weight in the dparam file.

Order of Precedence

The order of precedence is: bblink>linkdist>admincost>dparam.

bblink File: DIST, DISTA2Z, and DISTZ2A Fields

In the bblink file, distance can be specified in the bblink file Miscellaneous (MISC) field as

- DIST = x (applies to all trunks defined together, symmetric)
- DISTA2Z = x (asymmetric distance: distance from Node A to Node Z)
- DISTZ2A = x (asymmetric distance: distance from Node Z to Node A)

linkdist File

```
#from     to     dist
N01       N02    5
N01       N03    1
```

This file contains specific default distances (admin weights) that can be assigned between “from” and “to” backbone nodes. It overrides the admincost file. It can be overridden by a contradictory bblink file miscellaneous entry.

To generate a linkdist file based on actual link mileage, insert the line “linkdistunit=-1” in the linkdist file. Then read the linkdist file into your network and save your spec file. The newly created linkdist file will have a list of pairs of end points with the airline mileage between each pair of points used as the administrative weight between the two points.

Informational Note: This feature currently applies when the Routing Method (Tools > Options > Design, Path Placement tab) is set to “Admin. Weight”.

admincost File

New Format

```
#protocol hwtype1 hwtype2 tkType samesite samePG weight varWeight hlevel1 hlevel2 misc
ISIS      7600    7600    ALL    1        *      100    1        REGULAR  REGULAR  -1
```

Old Format

```
#protocol hwtype1 hwtype2 trunkType samesite samePeerGroup weight variableWeight
ADMINWEIGHT  ALL   ALL     ALL       *        *            100       0
ADMINWEIGHT  CORE  CORE    ALL      *        *            200       0
```

This file contains default distances (admin weights) that can be assigned between nodes of different hardware types and trunk types. It also contains a variable weight field that is multiplied by the airline distance between the link endpoints and added to the fixed weight field. It can be overridden by the default value between specific nodes in the linkdist file.

Field	Description
protocol	Routing protocol such as PNNI, OSPF, or ISIS for which the admin weight will apply
hwtype1 and hwtype2	Hardware types of link endpoints that must be matched for the rule to apply to that link.
tktype	Trunktype that must be matched for the rule to apply to a link
samesite	• 0: INTER (between nodes in two sites, or from a node not belonging to any user-defined site to a user-defined site) • 1: INTRA (between nodes in the same site) • 2: EXTRA (between nodes both of which are not in a site) • *: Don’t Care
samePG*	For PNNI models only. Allows user to have a different admin weight rule depending upon if the link’s endpoints are in the same peer group or not. • 0: False • 1: True • *: Don’t Care
weight and varWeight	Admin weight gets set by default to weight + linkmileage * varWeight. Make sure the routing method in the Design Options is set to Admin Weight.
hlevel1 and hlevel2	Hierarchical levels
misc	Your particular model may contain miscellaneous options.

dparam File: linkdistunit Parameter

If the admin weight for a link is not specified in the bblink file and no admin weight is specified between the node pair in the linkdist file, the admin weight will be determined by the linkdistunit parameter in the dparam file. This parameter can be specified in the "User-Defined Link Distance" function of the “Read Files” menu in bbdsgn.

- If linkdistunit is positive, that value is the default admin weight for all links.
- If linkdistunit is negative, the default admin weight for each link is calculated as the airline distance between the link endpoints divided by the absolute value of linkdistunit and rounded up to the nearest integer. (For example, suppose linkdistunit = -50. In that case, the user-defined distance for links with an airline distance between 1 and 50 miles is 1. Similarly, for links with an airline distance between 51 and 100 miles, the user-defined distance is set to 2.)

Fixing Links

Certain files and parameters can be used to keep links fixed.

fixlink File

The format of the fixlink file is exactly the same as that of the bblink file. This file is used to tell the program that the links specified in this file should not be deleted during backbone link design.

dparam File: maxlinkcheck, maxdivlinkcheck

The parameter maxlinkcheck sets the maximum number of links checked for deletion during a basic design.

The parameter maxdivlinkcheck sets the maximum number of links checked for deletion during a diversity design.

Node Constraints

nodeweight File

The nodeweight file can be used to assign: 1) Link Penalty for Design: a penalty at a node for purchasing links at it during a design, 2) Trunk-bandwidth-limit: a maximum trunk bandwidth capacity at a node that gets used during design, and 3) Transit-bandwidth-limit: a maximum transit demand bandwidth capacity for a node that gets used for path placement.

```
#nodeID/name  nodeweight trunk-bandwidth-limit transit-bandwidth-limit
N0007          100        500M                  1000M
N0011          -1         10808000
N0010          BLOCK
```

Link Penalty for Design (Originally Known as nodeweight)

Default value = 0.

Possible values: a number, NOPASS, or BLOCK.

1. The node_weight value in the nodeweight file is used to increase the trunk cost at a node by node_weight*average_cost_per_mile*0.3 during the design phase. This value is used to influence design only and is not reported in the trunk cost report.

2. If the node-weight value >=100000, is set to NOPASS, or BLOCK, it functions to prevent transit demands by marking the node as an end node:
   - If the node-weight value >= 100000 or is set to NOPASS, pass-through traffic is not allowed in *design* mode.
   - If the node-weight value is set to BLOCK instead of a numeric weight, pass-through traffic is not allowed through that node in both *design* and *simulation* modes.

Trunk-bandwidth-limit

Default value = infinity

The trunk-bandwidth-limit is used as a design constraint. New links are not added to a node if the bandwidth limitation will be exceeded.

For example, if the trunk-bandwidth-limit is set to 5M (5Mbps) and the link type being designed for is a T1, then only 3 T1s (each ~1.5 Mbps) can be added since 4 T1s would exceed this limit.

Note: A node weight is required if maximum link bandwidth capacity is to be specified. A node weight of -1 can be used as a place holder.

Two other parameters are related to the usage of this field: forcedesign and extratrunkpenalty. If forcedesign=1, and if extratrunkpenalty>=100, then cost of new trunks at the node is increased by extratrunkpenalty (unit= US dollar) in the trunk selection process. In other words, if we can justify a new trunk even if its cost is increased by extratrunkpenalty (dollar), then we'll add the new trunk ignoring the trunk-bandwidth-limit.

Transit-bandwidth-limit

Maximum transit demand bandwidth capacity is used as a path placement constraint. Demands are not placed through this node if the bandwidth limitation will be exceeded. Demands that originate or terminate at this node are not included in this bandwidth.

Reserved Bandwidth

Order of Precedence

rsvbwfile > dparam

dparam File: fixfat and fatpct Parameters

The fixfat and fatpct parameters in the dparam file are used to globally define reserved bandwidth for all links in the network.

rsvbwfile File

```
#node1      node2      fixfat      fatpct
N01         N02        128000      0.1
N05         N10        256000      0
```

To define reserved bandwidth for specific node pairs, the rsvbwfile should be used. Reserved bandwidth is calculated based on values in the rsvbwfile. For node pairs not defined in this file, reserved bandwidth is calculated based on the fixfat and fatpct global parameters. To follow the reservation constraints, bbdsgn will avoid using reserved bandwidth during path assignment and backbone design. In failure analysis/failure simulation routines, however, these reservation constraints are ignored.

Chapter 9

Output Files

Output files are the report files created by the bbdsgn program. These files are generated in the output directory that is specified in the File Manager, and are given the runcode extension defined in the specification file. This runcode enables you to distinguish the output file of one design from that of another. With the exception of the runcode extension, output filenames are specified by the bbdsgn program and may not be changed by the user.

The bbdsgn program creates both network information reports, and simulation and failure analysis reports. The runcode listed in the samples below is 'x', but the user may specify a different runcode for reports.

Report Menu

Here is a sampling of the different reports:

Menu Option	Report(s) Generated or Next Menu Level	Explanation or Other Usage Info
1. Path and Diversity	PATHRPT	Path and diversity report. See PATHRPT Report on page 88.
2. Link Bandwidth Allocations	LKBWRPT	Link bandwidth allocation report. See LKBWRPT Report on page 90.
	LKUTIL	Planned Link Utilization Report See LKUTIL Report on page 91.
3. Link Configuration and Cost	LKCOST	Link pricing report. See LKCOST Report on page 92.
	linkconf* (for Alcatel)	Link configuration parameter report. See linkconf Report on page 93.
4. Link Partition Information	LKPART (ATM or IP/MPLS)	Link Partition Report See LKPART Report on page 95.
5. Backbone Hardware Cost	BBHWCRPT	Backbone hardware cost report. See BBHWCRPT Report on page 96.
6. Demand Circuit Cost	CKTCOST	Demand Route Cost Report 1. Evaluate Demand Cost According to Circuit Routes 2. Evaluate End-to-End Connection Cost See CKTCOST Report on page 98.
7. Equivalent Path Report	EQPATHRPT (Router)	Equal Cost Multi-Path Report See EQPATHRPT Report on page 99.

Menu Option	Report(s) Generated or Next Menu Level	Explanation or Other Usage Info
8. International Price Report	INTLCOST CTRYCOST AVRGCOST	International link cost report. Country cost distribution report. Average link cost report. See INTLCOST Report on page 99, CTRYCOST Report on page 101, and AVRGCOST Report on page 102.
9. Load Report Based On Real Traffic	LINKLOAD (trafficload feature)	See LINKLOAD Report on page 103.
10. CoS Report (IP/MPLS only)		
	1. Demand CoS Report	DEMANDCOS See DEMANDCOS Report on page 105.
	2. Link CoS Report	LINKCOS See LINKCOS Report on page 106.
11. LSP Path Generation Report (IP/MPLS only. Requires tunnel license)		
12. BGP Report (IP/MPLS only)	BGPRPT	See BGPRPT Report on page 107.
13. Voice Report (Voice only)		
	1. Summary Report	VOICERPT
	2. Voice Trunk Group Report	TKGPRPT
	3. Voice Switch to Switch Traffic Statistics Report	
	4. Voice Path Report	
14. OSPF Report (Router)	OSPFSUMMARYRPT	

Other Reports Generated by the Report Manager

Report	Description
TUTIL (IP/MPLS)	Tunnel Utilization report.
INTDOMPATH (Domain feature)	Inter Domain Path report.
INTDOMLOAD (Domain feature)	Inter Domain Load report.
DVSIM	Link Diversity Utilization report.
DOMPASSTHRU (Domain or Area Feature)	Area Pass Through Paths report.
OSPFDETAILRPT (For OSPF (IP))	OSPF Area Detail report.
ABRBORDERRPT (For OSPF (IP))	ABR and its Bordering Area report.
InfUtil (IP/MPLS)	Interface Traffic report.
ASTRAFFIC (IP/MPLS)	AS Traffic report.
InterASTraffic (IP/MPLS)	Inter-AS Traffic report.
configLog (From Import Network)	Configuration Integrity Checks report.

Report	Description
ISISReport (IP/MPLS)	ISIS report.
VPNREPORT_LAYER3 (IP VPN)	Layer 3/Layer 2 Kompella VPN report.
VPNREPORT_LAYER2 (IP VPN)	Layer 2 Martini VPN report.
SWITCHCONN	Switch Connections Statistics report.
TUNNELRPT (IP/MPLS)	Tunnel Path & Diversity report.
TUNNEL_LOAD (IP/MPLS)	Measured Tunnel Traffic report.
PATHBW (ATM)	Equivalent Capacity report.

Simulation and Other Reports

Report	Description
MAXLKUTIL	Peak link utilization report. See MAXLKUTIL Report on page 108
SIMRPT	Interactive simulation report. See SIMRPT Report on page 109
LKFAIL	Link failure simulation report. See LKFAIL.x Report on page 111
LINEFAIL	Single line failure simulation report. See LINEFAIL.x Report on page 113
NDFAIL	Node failure simulation report. See NDFAIL.x Report on page 114.
FACFAIL	Facility Failure report. See FACFAIL.x Report on page 116.
DAILYFAIL	Daily random failure simulation report. See DAILYFAIL.x Report on page 117.
DAILYSEQ	Daily random failure sequence. See DAILYSEQ.x Report on page 118.
RNDFAIL	Random failure simulation report for up-down sequences.
RNDLKUTL	Random link utilization report for up-down sequences. See RNDLKUTL.x Report on page 121.
RNDPATH	Random path placement report for up-down sequences. See RNDPATH.x Report on page 121.
SIMPLACE	Path placement simulation report. See SIMPLACE.x Report on page 122.
TRAFFICLOAD	Link load report See TRAFFICLOAD.x Report on page 122.
PATHDELAY	Path delay information report. See PATHDELAY.x Report on page 123.
PeakSimSummary	Peak Simulation Summary report. See <Link>PeakSimSummary.x Report on page 9-124.
PeakSimLink	Link oversubscription from Peak Simulation. See <Link>PeakSimLink.x Report on page 9-125.
PeakSimRoute	Failed demands and tunnels from Peak Simulation. See <Link>PeakSimRoute.x Report on page 9-125.

PATHRPT Report

The Path and Diversity Report, PATHRPT.x, shows the actual paths routed by the bbdsgn program. It also indicates the diversity level supported by the diversity groups. Site diversity is satisfied if the two paths in the same diversity group pair route through site disjoint paths. Link diversity level is satisfied if the two paths in the same diversity group pair route through disjoint links. They may pass through the same intermediate nodes.

The following sections show a sample of the PATHRPT report and an explanation.

Sample PATHRPT.x Report

```
************************************************************************
*
*    PATH INFO by TG/Diversity Group -- runcode=x
*
************************************************************************
*  Notations:
*         & : Same site
*         - : Intra-LATA or Intra-Country
*        -- : Inter-LATA or Inter-Country
*         = : Intra-LATA or Intra-Country, Second vendor or linktype
*        == : Inter-LATA or Inter-Country, Second vendor or linktype
*         @ : Intra-LATA or Intra-Country, Inter-domain
*        @@ : Inter-LATA or Inter-country, Inter-domain
*       -*- : Link missing or not enough bandwidth
*         ~ : Secondary Priority/Preempt Priority used
*        ~~ : Secondary Priority/Preempt Priority used
*        Rn : Data, n=number of demands (default=1)
*        Vn : Voice, n=number of demands (default=1)
*       VDn : Demand Voice, n=number of demands
*      Dxxx : xxx - Name of diversity group
*      Ixxx : xxx - Database record ID
*        PR : Path Required (will not be rerouted)
*        PS : Path Select
*        BT : T - Transparent signaling
*        BP : P - Pass-through timing
*        BA : A - Asynchronous
*     BCsize: C - Committed burst size (unit=bit)
*     BEsize: E - Excess burst size (unit=bit)
*     BPsize: P - Portspeed or peak rate (unit=bit)
*      BDsec: D - Max delay of path (unit=sec)
*       Cxxx: xxx - Card type
*      Uyyy : yyy=Utilization percentage (0.1 means 10%)
*        A2Z: One-way circuit, traffic flows from FromNode to ToNode
*        Z2A: One-way circuit, traffic flows from ToNode to FromNode
*     #! msg: msg=Comments specified by the user

*     Maximum number of links allowed in a path=8
*
*  1464 demand requirements, 1464 routed
* Average number of links in demand paths = 1.67
*
* Bandwidth Unit = bit
*
*CircuitID From_Node  To_Node       Speed  Type Priority Path_Spec  Comment
*
ckt12      N01        N02           56000  R 10,10 N1--N2
ckt43      N01        N05           32000  V 07,07 N1--N5
ckt42      N01        N05           32000  V 07,07 N1--N5
```

```
ckt51       N01        N06           56000  R 10,10 N1--N2--N6
ckt47       N01        N06           32000  V 07,07 N1--N2--N6
```

Explanation: PATHRPT.x

The runcode you specified in your specification file is indicated in the title line of the PATHRPT report in place of the "x" shown in the sample report.

The bbdsgn program uses the following conventions and notations in describing the paths in the network:

Symbol	Example	Explanation of Example
& (ampersand)	N79&N66	An ampersand connecting two nodes describes a path from N79 to N66, but it also indicates that the two nodes belong to the same site.
- (dash)	N1-N2-N3	A dash connecting two nodes describes a path from N1 to N2 to N3 with two hops. The dash also indicates that the two nodes it connects are in the same LATA (Intra-LATA) or country (Intra-Country).
-- (double dash)	N7--N79	A double dash connecting two nodes describes a path from N7 to N79. It indicates that the two nodes are in different LATAs (Inter-LATA) or countries (Inter-Country).
= (equal)	N72=N61	An equal sign connecting two nodes describes a path from N72 to N61. The equal sign indicates that this is an Intra-LATA or Intra-Country link using either the second vendor or linktype.
== (double equal)	N2==N15	A double equal sign connecting two nodes describes a path from N2 to N15. The double equal sign indicates that this is an Inter-LATA or Inter-Country link using either the second vendor or linktype.
@ (at)	N7@N8	An at symbol connecting two nodes describes a path from N7 to N8. The at symbol indicates that this is an Intra-LATA or Intra-Country link of an Inter-domain call.
@@ (double at)	N14@@N33	A double at symbol connecting two nodes describes a path from N14 to N33. The double at symbol indicates that this is an Inter-LATA or Inter-Country link of an Inter-domain call.
~ (tilde)	N96~N97	A tilde symbol connecting two nodes describes a path from N96 to N97. The tilde symbol indicates that this is an Intra-LATA or Intra-Country link using secondary priority/preempt. This notation is only for net.com hardware devices.
~~ (double tilde)	N99~~N100	A double tilde symbol connecting two nodes using secondary priority/preempt and its inter-Lata or inter-country link.

Keeping these conventions in mind, the path specification N7--N79&N66 is equivalent to N7-N79-N66.

Except that the first specification gives you more information: N7--N79 indicates that N7 and N79 are in different LATA and N79&N66 indicates that N79 and N66 belong to the same site.

Note that a link is defined as a group of private lines connecting the same pair of nodes, leased from the same carrier, and having the same bandwidth types. If there are multiple links between the same pair of nodes, then the equal sign (=) or the double equal sign (==) are used to indicate the second links.

If the links between two nodes are bought from more than one vendor, and the links are not in the same site, then use - (for Intra-LATA) or -- (Inter-LATA) to indicate the link from the first vendor, and = (Intra-LATA) or == (Inter-LATA) to indicate the other link vendors.

For example, the following two paths

- N1--N2
- N1==N2

satisfy link diversity because they are carried on different vendors' links. If N1 and N2 are the two end nodes, then they are considered to be site diverse paths.

The possible type fields in the circuit paths are explained in the rest of the notation section. The meaning of the circuit path fields can be found in the section describing the demand file.

Unplaced circuits are marked with the notation -*- to signify a missing link in the path, or insufficient bandwidth.

In the above example, there are 1464 circuit requirements in the demand files. All of them are routed. The average number of links in the circuit paths is 1.67. The maximum number of links allowed in a path is 8.

Circuits in transmission groups and diversity groups are reported before other circuits. The section:

```
* TG I002044 - Site diversity
I002044A  N67  N18  512000  R  11/11  N67-N18
I002044B  N77  N18  512000  R   7/7   N77-N18
```

means that both of the circuits I002044A and I002044B belong to the Transmission Group I002044. The two paths, N67-N18 and N77-N18 satisfy site diversity. Other possible values for the "Site diversity" comment field are Link diversity and No diversity.

If there are multiple circuits in a diversity group, then they are grouped in pairs.

LKBWRPT Report

The Link Bandwidth Allocation Report, LKBWRPT.x, shows information on each link’s bandwidth capacity, used bandwidth, available bandwidth and reserved bandwidth. The following sections show a sample of the link bandwidth report and an explanation.

Sample LKBWRPT.x Report

```
******************************************************************************
*
*    LINK BANDWIDTH REPORT -- runcode=xx
*
******************************************************************************

N1        =X001           N2        =X002
N3        =X003           N4        =X004
N5        =X005           N6        =X006
N7        =X007           N8        =X008
…
*******************************************************************************
Notations:
      MaxCap  = Total bandwidth - Link overhead
      DataBw  = Bandwidth allocated for circuit requirements
               Per circuit overhead required by hardware is included
      AvailBw = MaxCap - DataBw
      RsvBw   = amount of available bandwidth that is
                reserved for future use.
      Bandwidth Unit = bit

From       To        Vendor       Type   #    MaxCap   DataBw   AvailBw(  RsvBw)
N1         N2          ATT          T1   1     1.484M   1.439M  44.800K(      0)
N1         N2          USS          T1   1     1.484M 672.000K 812.000K(      0)
N1         N3          ATT          T1   1     1.484M   1.045M 439.200K(      0)
N1         N4          WTG          T1   1     1.484M 681.200K 802.800K(      0)
N1         N5          ATT          T1   1     1.484M 899.600K 584.400K(      0)
...
```

Explanation: LKBWRPT.x Report

The runcode you specified in your specification file is indicated in the title line of the Link Bandwidth report in place of the "x" shown in the sample report.

This report begins by listing the ID and name of each node in the network. Following this list are some notations defining terms used in the report. The RsvBW field uses the reserved bandwidth specified in the rsvbwfile input file.

In the above example, the total bandwidth available between nodes 1 and 4 for data/voice circuits is 1,484 Kb. The amount of bandwidth used is 681.200 Kb. The amount of bandwidth available to other circuits is 802.800 Kb.

Since the reserved bandwidth is not a constraint set at the hardware, the reserved bandwidth are used by bbdsgn for path placement while doing the simulation and network failure analysis.

LKUTIL Report

The Link Bandwidth Utilization Report, LKUTIL.x, shows information on each link's bandwidth utilization--used bandwidth, available bandwidth, and any overhead that is used.

Sample LKUTIL.x Report

The following sections show a sample of the link bandwidth utilization report.

```
#########################################################################
*
*     LINK UTILIZATION REPORT     runcode=atm
*
*
#########################################################################
#    AvailBw = available bandwidth in the link
#    UsedBw = bandwidth used by demands
#    Ovhd = link overhead
#    TotalBw = AvailBw + UsedBw + Ovhd
#    Bandwidth Unit = bit
#
Linkname                 Type       TotalBw  AvailBw   UsedBw     Ovhd
N1->N2                          T1   1.536M 681.216K 854.784K        0
N1<-N2                        ----   1.536M 681.216K 854.784K        0
N1->N2                          T1   1.536M   6.208K   1.530M        0
N1<-N2                        ----   1.536M   6.208K   1.530M        0
N1->N3                          T1   1.536M 278.416K   1.258M        0
N1<-N3                        ----   1.536M 278.416K   1.258M        0
N1->N4                          T1   1.536M 805.448K 730.552K        0
```

```
N1<-N4                        ----   1.536M 805.448K 730.552K        0
N1->N5                          T1   1.536M 604.048K 931.952K        0
N1<-N5                        ----   1.536M 604.048K 931.952K        0
```

LKUTIL report csv format has been modified.

Old Format: lkname,bwtype,...

New Format: lkname,node1,node2,bwtype,...

LKCOST Report

The Backbone Link Configuration and Pricing Report, LKCOST.x, summarizes the total number of circuits terminating at each node. It also gives detailed pricing information for each link. The following sections show a sample of the LKCOST report and an explanation.

Sample LKCOST Report

```
******************************************************************
*    Backbone Link Configuration and Pricing Report    -- runcode=xx
*
*    Requirements:
*                   670 two-way data  demands, total bandwidth=   17.319M bps
*                   794 two-way voice demands, total bandwidth=   25.536M bps
*                    52            T1:cost=$    290,116.87/mon
*                     1          F-64:cost=$          0.00/mon
*                     4         F-128:cost=$          0.00/mon
*                    14         F-256:cost=$     24,360.39/mon
*                    13         F-384:cost=$    164,186.52/mon
*                    37         F-512:cost=$    432,239.95/mon
*                    13         F-768:cost=$     59,138.37/mon
                  Total link cost=$    970,042.11/mon
*                     Average number of hops = 1.67
*                     Average link utilization pct = 65.68%
*                     Average link fat reservation pct = 0.00%
*
*                     Currency= DL(American Dollar)
*                     Date= 3/9/99 11:07
*
******************************************************************
demand file: /e/wandl/example/domain.intl/demand.xx
* Parameter Values
* T1bw: 1544000, T1lkovhd: 60000,  E1bw: 2048000, E1lkovhd: 20000,
* T3bw: 44736000, T3lkovhd: 5888000
* TAovhd: 24000, TAovhdpp: 2000, TAbwreq: 8000
* fatpct: 0.00%, fixfat: 0
* hopdelay: 100, maxhop: 8, phyhoplimit: 12
* hopdist: Actual_Mileage

N1      (X001        ):  17 links,     N2      (X002        ):   7 links,
N3      (X003        ):   5 links,     N4      (X004        ):   8 links,
N5      (X005        ):   2 links,     N6      (X006        ):   5 links,
N7      (X007        ):   7 links,     N8      (X008        ):  10 links,
N9      (X009        ):  10 links,     N10     (X010        ):   4 links,
N11     (X011        ):   6 links,     N12     (X012        ):   7 links,

**** Detailed Pricing Information

----- Source -----   ------ Dest ------  Vendor  #       Link   Monthly    NRC
919878 N2            919878 N53           NET     1         T1         0        0
```

```
212719 N3            212613 N4            LEC     2         T1      972      816
212719 N3            718917 N48           LEC     1         T1      772      408
212613 N4            718917 N48           LEC     1     FT256K      772      408
…
----------------------------------------------------------------------------
202268 N1            919878 N2            ATT     1         T1     6201     2846
202268 N1            919878 N2            USS     1         T1     6193     3398
202268 N1            212719 N3            ATT     1         T1     6057     1520
202268 N1            212613 N4            WTG     1         T1     3839     1310
202268 N1            717821 N5            ATT     1         T1     6296     1114
202268 N1            901722 N7            USS     1     FT512K     5178     3398
202268 N1            312765 N8            ATT     1         T1     7545     1638
…
--------------------------------------------------------------------------------
TOTAL                                           134             970042  281835
Total 134 links, monthly= $ 970,042, nrc=$ 281,835
(42 links can not be priced)
```

Explanation: LKCOST.x Report

The LKCOST.x report contains reference as well as summary information:

- The top portion of the LKCOST report, marked with asterisks, provides summary information about the configuration used in this network.
- The next line gives the path and name of the datapath file, followed by some lines describing the parameter values used. This section provides a handy reference to remind you of the values used in generating this design
- The next segment lists the nodes and the number of links associated with each node.
- The Detailed Pricing Information segment lists each link sorted by vendor, with its monthly cost and total non-recurring charges. The final line provides the total number of links, the total monthly cost, and the total non-recurring charges for the network as currently configured.

linkconf Report

The Link Configuration Parameter Report, linkconf.x, provides an exhaustive list of detailed information of each link's configuration. The following sections show a sample of the linkconf report and an explanation.

Sample linkconf Report

This report may also be viewed using the **Report Viewer** tool.

```
##################################################
#   Link Configuration Parameter Report
##################################################
#  Software Release= 3.4.0,  Compilation Date= 20020918
#  Report Date= 9/18/2002 10:48   Runcode=jk1  User=wandl

LinkName,NodeA,PortA,HwtypeA,NodeZ,PortZ,HwtypeZ,Vendor,BwType,Status,MinPri,MaxPri,OverBF,ReBFac,5620-A
WA2Z,5620-AWZ2A,PNNIBW,PNNI-AWA2Z,PNNI-AWZ2A,RCCbw,SignalBW,PNNI-OverBF,MaxPNNICNT,AggrToken,MaxCkt,
,A.1.1,,7470,A.1.2,,7470,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=28.000000M,50
40,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.1.2,,7470,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=149.760000M,5
040,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.1.2,,7470,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=149.760000M,5
040,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.2.1,,7470,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=86.000000M,50
40,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.2.3,,7670,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=28.000000M,50
40,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.2.3,,7670,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=38.000000M,50
40,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.2.3,,7670,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=149.760000M,5
040,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
,A.1.1,,7470,A.3.2,,7470,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.00,100,100,PNNIBW=149.760000M,5
040,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.000;ABR:4.000;,,AGGR=none,,
```

Explanation: linkconf.x Report

The linkconf.x report contains detailed configuration information for each backbone link in the current network. Every link entry in the linkconf file contains 25 fields, each separated by a (,) comma. The fields are as follows:

LinkName, NodeA, PortA, HwtypeA, NodeZ, PortZ, HwtypeZ, Vendor, BwType, Status, MinPri, MaxPri, OverBF, ReBFac, 5620-AWA2Z, 5620-AWZ2A, PNNIBW, PNNI-AWA2Z, PNNI-AWZ2A, RCCbw, SignalBW, PNNI-OverBF, MaxPNNICNT, AggrToken, MaxCkt

Consider the first link entry in the example below:

```
link01,A.1.1,,7470,A.1.2,,7470,BTE,STM1,,MinPri=16,MaxPri=1,BFAC=1.00,ReBFAC=1.0
0,100,100,PNNIBW=28.000000M,5040,5040,RCCbw=0,SIGOVHD=0,PNNI_OVF=RT:2.000;NRT:4.
000;ABR:4.000;,,AGGR=none,,
```

Position	Field	Corresponding Value for Sample Entry
1	LinkName	LINK01
2	NodeA	A.1.1
3	PortA	
4	HwtypeA	7470
5	NodeZ	A.1.2
6	PortZ	
7	HwtypeZ	7470
8	Vendor	BTE
9	BwType	STM1
10	Status	
11	MinPri	16
12	MaxPri	1
13	OverBF	1.00
14	ReBFac	1.00
15	5620-AWA2Z	100
16	5620-AWZ2A	100
17	PNNIBW	28.000000M
18	PNNI-AWA2Z	5040
19	PNNI-AWZ2A	5040
20	RCCbw	0
21	SignalBW	0
22	PNNI-OverBF	RT:2.000;NRT:4.000;ABR:4.000
23	MaxPNNICNT	

Position	Field	Corresponding Value for Sample Entry
24	AggrToken	NONE
25	MaxCkt	

LKPART Report

The Link Tunnel Partition Information Report, LKPART.x, (available for ATM networks) displays the over subscription factor and detailed link partition information for each link in the network. This report displays the same detailed information in the capacity tab menu of the Link Info window found in the Network menu.

Sample Link Tunnel Partition Information

```
*************************************************************************
*
*    Link Tunnel Partition Information -- runcode=mpls-fish
*
*************************************************************************
##  Software Release= 3.5.2,  Compilation Date= 20030305
##  Report Date= 3/6/2003 12:24   Runcode=mpls-fish  User=wandl

*   IGP   #Flow= Number of layer 3 flows routed through
*               the link NOT following tunnel paths
*   IGP  FlowBW= Total Bandwidth of layer 3 flows routed through
*               the link NOT following tunnel paths
*   TE #Flow= Number of layer 3 flows routed through
*               the link following TE tunnel paths
*   TE FlowBW= Total Bandwidth of layer 3 flows routed through
*               the link following TE tunnel paths
*   TE RSVP-BW = RSVP bandwidth configured for TE tunnels
*   TE AvRSVP= RSVP bandwidth available for additional tunnels


LINK1      ATL        HOU        ATT         1         OC3
*** ATL->HOU ***
Partition    #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP             9  22.807M        0        0        0
GlbPool         2   1.275M   1.000M 155.520M 154.520M
SubPool         0        0        0        0        0
*** HOU->ATL ***
Partition    #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP             7   3.719M        0        0        0
TE              0        0        0 155.520M 155.520M
TE              0        0        0        0        0
----------------------------------------------------------------------------

LINK18     ATL        LAX        ATT         1         OC3
*** ATL->LAX ***
Partition    #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP            20  71.980M        0        0        0
GlbPool         0        0        0 155.520M 155.520M
SubPool         0        0        0        0        0
*** LAX->ATL ***
Partition    #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP            20 144.240M        0        0        0
TE              0        0        0 155.520M 155.520M
TE              0        0        0        0        0
----------------------------------------------------------------------------
```

```
LINK2      ATL        WDC        ATT         1         OC3
*** ATL->WDC ***
Partition   #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP            26 176.420M        0        0        0
GlbPool         0        0        0 155.520M 155.520M
SubPool         0        0        0        0        0
*** WDC->ATL ***
Partition   #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP            30 122.305M        0        0        0
TE              0        0        0 155.520M 155.520M
TE              0        0        0        0        0
---------------------------------------------------------------------------
LINK3      BOS        DET        ATT         1         OC3
*** BOS->DET ***
Partition   #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP             8  13.771M        0        0        0
GlbPool         9  52.547M  10.000M 155.520M 145.520M
SubPool         0        0        0        0        0
*** DET->BOS ***
Partition   #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP             8  26.493M        0        0        0
TE              9  52.924M  15.000M 155.520M 140.520M
TE              0        0        0        0        0
---------------------------------------------------------------------------

LINK4      BOS        NYC        ATT         1         OC3
*** BOS->NYC ***
Partition   #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP             4  35.683M        0        0        0
GlbPool         0        0        0 155.520M 155.520M
SubPool         0        0        0        0        0
*** NYC->BOS ***
Partition   #flow   FlowBW TunnelBW  RSVP-BW  AvRSVP
IGP             4  20.188M        0        0        0
TE              0        0        0 155.520M 155.520M
TE              0        0        0        0        0
---------------------------------------------------------------------------
```

BBHWCRPT Report

The purpose of the Backbone Hardware Cost Report, BBHWCRPT.x, is to provide information on link usage and hardware facility usage costs to ease the comparison of different designs, and to determine the cost for specific applications.

Backbone Link Cost-Per-Month Segment

The following sections show a sample of the Backbone Link Cost Per Month segment of the BBHWCRPT.x report as well as an explanation.

Sample Backbone Link Cost-Per-Month Segment

```
****** BACKBONE LINK COST PER MONTH *******

INHOUSE:

                                                                   Link
From       To         Vendor  #         Type  LinkCost   Usage%  UsgCost
N01        N27         NET    1           T1          0   17.68%        0
```

```
N04        N06         NET    3           T1         0   31.27%        0
...
N67        N77         NET    1           T1         0   39.57%        0
N69        N79         NET    1           T1         0   79.78%        0
--------------------------------------------------------------------------------
TOTAL                        46                      0   44.53%        0

FIBER:  none

LEC:

                                                                   Link
From       To         Vendor  #         Type  LinkCost   Usage%  UsgCost
N01        N24         LEC    1           T1   $582.88   56.28%  $328.05
N01        N45         LEC    1           T1  $2351.37   38.81%  $912.66
...
N71        N105        LEC    1           T1   $878.27   30.19%  $265.14
N88        N116        LEC    1           T1   $789.39   37.74%  $297.88
--------------------------------------------------------------------------------
TOTAL                        64               58923.76   49.02%  $28884.16

IXC:

                                                                    Link
From       To         Vendor  #         Type  LinkCost   Usage%   UsgCost
N01        N18         USS    2           T1  $2771.05   90.03%  $2494.78
N01        N22         USS    1           T1  $3645.93   92.83%  $3384.52
…
N76        N82         MCI    1           T1  $4683.82   43.13%  $2019.98
N79        N92         MCI    1           T1  $4806.42   30.19%  $1450.99
N82        N92         USS    1       FT512K  $3525.62   38.81%  $1368.43
--------------------------------------------------------------------------------
TOTAL                       141              526193.63   57.04%  $300150.97

Total Link Cost   = $ 585117.38/mon
Total Link Usage Cost   = $ 337612.16/mon
```

Explanation: Backbone Link Cost-Per-Month Segment

The following column headings are used in the Backbone Link Cost Per Month segment of the BBHWCRPT.x report:

Field	Description
From, To, Vendor, and Type	These fields provide the relevant information needed to calculate the price of a link.
#	This field shows the number of circuits.
LinkCost	This field lists the total cost for the circuits specified in the # field.
Usage%	This field displays the percentage of the total available bandwidth being utilized.
LinkUsgCost	The link usage cost field shows the cost of the link according to the utilization percentage.

For example, in the IXC portion of the sample report shown above, there are 2 Sprint T1 links from node 1 to node 18. The cost of both T1s is $2771.05 per month. The node is currently utilizing 90.03% of the available bandwidth. The cost of the portion utilized is $2494.78 per month (the product of $2771.05 and 90.03%).

CKTCOST Report

The Circuit Cost Report, CKTCOST.x, displays the cost per demand between two nodes. Demand costs may be determined by either of two methods:

- Evaluate Demand Cost According to Circuit Routes
- Evaluate End-to-End Connection Cost

Demand cost according to circuit routes is determined by the following method:

On the other hand, demand cost according to end-to-end connection cost is determined by calculating the circuit cost between the origination and destination node pair. The cost calculated is not dependent on the actual routing of the circuit through the backbone network.

The following is an example output of demand cost according to circuit routes:

```
##############################################################
#   COST REPORT FOR CIRCUIT DEMANDS             runcode=xx
##############################################################

#                     Currency= DL(American Dollar), DistUnit=mile
U.S. Pricing option:
     1. Default IXC vendor: Least cost,
     2. Default intra-lata vendor: LEC,     3. Jurisdiction: INTER-STATE

   Ratedir = /u/wandl/db/rates/default
# Calculation Method: utilization cost according to circuit routes
#   i.e.  circuit cost = Sum of (bandwidth/link_cap)*(link cost)
#                        for links in the path
#
#CircuitID   Node        Node     Bandwidth  #ckt dir  Mile #hop   Cost/ckt

ckt12      N01         N02          56.000K R   1 <->   229    1 $    234.01
ckt23      N01         N02          56.000K R   1 <->   229    1 $    234.01
ckt21      N01         N02          56.000K R   1 <->   229    1 $    234.01
ckt20      N01         N02          56.000K R   1 <->   229    1 $    234.01
ckt11      N01         N02          56.000K R   1 <->   229    1 $    234.01

The following is an example output of demand cost according to end-to-end
connection costs:

##############################################################
#   COST REPORT FOR CIRCUIT DEMANDS             runcode=xx
##############################################################

#                     Currency= DL(American Dollar), DistUnit=mile
U.S. Pricing option:
     1. Default IXC vendor: Least cost,
     2. Default intra-lata vendor: LEC,     3. Jurisdiction: INTER-STATE

   Ratedir = /u/wandl/db/rates/default

End-to-End Demand Cost Evaluation Option
   1. Voice Circuits: Regular VG Tariff
   2. Low Speed Data Circuits (<=64000): Regular DDS/FT1 Tariff
   3. High Speed Data Circuits (>64000): Regular FT1 Tariff
#
#CircuitID   Node     Node     Bandwidth  #ckt dir  Mile Tariff        Cost/ckt
```

```
ckt12      N01        N02          56.000K R   1 <->   229  USS  FT56K $
748.28
ckt23      N01        N02          56.000K R   1 <->   229  USS  FT56K $
748.28
ckt21      N01        N02          56.000K R   1 <->   229  USS  FT56K $
748.28
ckt20      N01        N02          56.000K R   1 <->   229  USS  FT56K $
748.28
ckt11      N01        N02          56.000K R   1 <->   229  USS  FT56K $
748.28
```

EQPATHRPT Report

The Equivalent Path Bandwidth Report, EQPATHRPT.x, generates information regarding the equivalent bandwidth for each demand found in the network. The equivalent path bandwidth is calculated using the standard formula for equivalent capacity based on Peak/Mean/Burst, CLR (cell loss ratio), and trunk buffer size. Since over subscription can be set on a per trunk basis, the over-subscription factor is not considered.

Sample Equivalent Path Bandwidth Report

```
************************************************************************
*    PATH BANDWIDTH INFORMATION -- generic
************************************************************************
#  Software Release= 3.2.0,  Compilation Date= 20020522
#  Report Date= 5/22/2002 11:12   Runcode=generic
CAC Calculation Method: EQcap+Gaussian
Parameters Used in Equivalent Capacity Calculation:
   Buffer_Size Unit= Byte
QoS   Overflow_probability   Buffer_Size
CBR                    1e-08      33.920K
RT                     1e-08      33.920K
NRT                    1e-06      33.920K
UBR/ABR                1e-06     530.000K

* Bandwidth Unit = bit
Note: Peak, Mean, and BurstSz are displayed in terms of Payload
        Equivalent Capacities are displayed in terms of Cells

Name    From    To        QoS Dir       BW     Mean     Peak  BurstSz   Eq_Cap
Demand1 N1      N2        CBR <-> 900.000K 900.000K 900.000K 900.000K 993.856K
Demand2 N2      N3        CBR <-> 750.000K 750.000K 750.000K 750.000K 828.496K
RN4N2   N4      N2        CBR <->  50.000K  50.000K  50.000K  50.000K  55.544K
Demand5 N1      N4        CBR <-> 150.000K 150.000K 150.000K 150.000K 165.784K
```

INTLCOST Report

The International Link Cost Report, INTLCOST.x, summarizes all of the international backbone links in the network in terms of bandwidth and utilization. It also gives detailed pricing information for each link. The following section shows a sample of the INTLCOST report and an explanation.

Sample INTLCOST Report

```
******************************************************************
*
```

```
* INTERNATIONAL LINK COST REPORT
*                            Currency= DL(American Dollar)
*                                Date= 3/9/99 11:13
*
*        Note: Peak Utilization Simulation is not performed
******************************************************************
*  Notations:
*       Type field:
*                    link - link cost specifed in bblink file or linkcost file
*                    n-n - cost defined in usercost file
*                    c-c - cost calculated from countrycost file
*                    est - cost estimated from user defined tariff
*                    db - cost calculated from tariff database
*                    ? - pricing failed
*       A2Zcost: half channel cost from Loc A to Loc Z
*       Z2Acost: half channel cost from Loc Z to Loc A
*
*       A2Zcost and Z2Acost are set to 0 if
*        1. Loc A and Loc Z are in the same country
*        2. pricing failed
*        3. price is obtained or estimated from usercost file
*
LinkName,Aloc,Card,Zloc,Card,Acountry,Zcountry,Vendor,AVendor,ZVendor,type,RawKbits,Kbits,A2Zcost,Z2ACost
,TotalCost,Flag,Util%,UtilC
ost,
LINK1,D100N1,C21,D100N52,C2,UK,PO,BTPC,BTEL,PCTT,FT256K,256,236,6952.09,7311.79,14263.88,db,0.000000,0.00
,
LINK2,D100N1,,D100N64,,UK,UK,NET,,,F-1984,2048,1964,0.00,0.00,0.00,db,0.000000,0.00,
LINK4,D100N1,C2,D100N242,C48,UK,UK,MER,,,F-1984,2048,1964,0.00,0.00,2357.79,db,0.000000,0.00,
LINK5,D100N3,C7,D100N238,C35,UK,BE,MEBE,MERC,BELG,12/12*T1,768,749,22107.63,12960.36,35067.99,db,0.854473
,29964.63,
LINK3,D100N5,C7,D100N168,C61,EI,EI,TLE,,,8/8*F-1920,512,493,0.00,0.00,29649.14,db,0.778905,23093.85,
LINK12,D100N6,C7,D100N98,C82,SZ,SZ,SPT,,,8/8*F-1920,512,493,0.00,0.00,65272.89,db,0.908722,59314.92,
LINK11,D100N9,C6,D100N203,C19,UK,UK,MER,,,F-1984,2048,1964,0.00,0.00,55571.36,db,0.293279,16297.91,
LINK13,D100N9,C7,D100N203,C117,UK,UK,MER,,,F-1984,2048,1964,0.00,0.00,55571.36,db,0.000000,0.00,
LINK15,D100N10,,D100N23,,SP,SP,TEL,,,F-1984,2048,1964,0.00,0.00,27816.34,db,0.000000,0.00,
LINK16,D100N10,,D100N23,,SP,SP,TEL,,,F-1984,2048,1964,0.00,0.00,27816.34,db,0.000000,0.00,
LINK17,D100N11,C7,D100N177,C57,GE,GE,DBP,,,F-1920,1920,1900,0.00,0.00,1237.28,db,0.000000,0.00,
LINK20,D100N15,C7,D100N174,C26,GE,GE,DBP,,,F-1920,1920,1900,0.00,0.00,55878.09,db,0.842105,47055.23,
```

Explanation: INTLCOST.x Report

The INTLCOST.x report contains detailed information for each backbone link in the current network. Every link entry in the INTLCOST file contains 19 fields, each separated by a (,) comma. The fields are as follows:

LinkName, Aloc, Card, Zloc, Card, Acountry, Zcountry, Vendor, Avendor, Zvendor, Type, RawKbits, Kbits, A2Zcost, Z2Acost, TotalCost, Flag, Util%, UtilCost.

Consider the first link entry in the example above:

```
LINK1,D100N1,C21,D100N52,C2,UK,PO,BTPC,BTEL,PCTT,FT256K,256,236,6952.09,7311.79,
14263.88,db,0.000000,0.00,
```

Position	Field	Corresponding Value for Sample Entry
1	LinkName	LINK1
2	Aloc	D100N1
3	Card	C21

Position	Field	Corresponding Value for Sample Entry
4	Zloc	D100N52
5	Card	C2
6	Acountry	UK
7	Zcountry	PO
8	Vendor	BTPC
9	Avendor	BTEL:
10	Zvendor	PCTT
11	Type	FT256K
12	RawKbits	256
13	Kbits	236
14	A2Zcost	6952.09
15	Z2Acost	7311.79
16	TotalCost	14263.88
17	Flag	Db
18	Util%	0
19	UtilCost	0

A FT256 link (LINK1) is represented between D100N1C21 and D100N52C2. N1 is located in the UK (United Kingdom), and N2 is located in PO (Portugal). The carrier is British Telecom on the UK side, and PCTT on the Portugal side. In terms of raw bandwidth, 256 Kb is allocated, but only 236 Kb is available after considering overhead. The cost from the UK to PO is 6952.09, and the cost from PO to UK is 7311.79 (reflecting differences in country pricing).

CTRYCOST Report

The Country Cost Distribution Report, CTRYCOST.x, shows the distribution of links on a per country basis in terms of Intra-Country and Inter-Country. It also provides information for each link on cost and utilization. The following section shows a sample of the CTRYCOST report and an explanation.

Sample CTRYCOST Report

```
Country,name,#intra,intra-cost,#inter,inter-cost, TotalCost, UtilCost,
AU,AUSTRALIA,2,0.00,5,39889.07,39889.07,15566.47,
BE,BELGIUM,0,0.00,3,0.00,0.00,0.00,
BR,BRAZIL,0,0.00,1,0.00,0.00,0.00,
CH,CHINA,1,0.00,3,0.00,0.00,0.00,
CO,COLUMBIA,0,0.00,2,24000.00,24000.00,16879.12,
DK,DENMARK,0,0.00,2,0.00,0.00,0.00,
FR,FRANCE,0,0.00,6,7134.00,7134.00,4640.00,
GE,GERMAN_FED,0,0.00,2,0.00,0.00,0.00,
HK,HONG_KONG,0,0.00,16,93597.50,93597.50,30024.37,
IN,INDIA,0,0.00,1,0.00,0.00,0.00,
EI,IRELAND,0,0.00,2,0.00,0.00,0.00,
IT,ITALY,0,0.00,2,0.00,0.00,0.00,
```

Explanation: CTRYCOST.x Report

The CTRYCOST.x report contains detailed information regarding link cost and utilization on a per country basis. Every link entry in the CTRYCOST file contains 8 fields, each separated by a (,) comma. The fields are as follows:

Country,Name,# Intra-LATA/Country Circuits,Intra-LATA/Country Cost,# Inter-LATA/Country Circuits,Inter-LATA/Country Cost,Total Cost,Utilization Cost

In the first link entry in the example above:

```
AU,AUSTRALIA,2,0.00,5,39889.07,39889.07,15566.47,
```

Position	Field	Corresponding Value for Sample Entry
1	Country	AU
2	Name	Australia
3	# Intra-LATA/Country Circuits	2
4	Intra-LATA/Country Cost	0.00
5	# Inter-LATA/Country Circuits	5
6	Inter-LATA/Country Cost	39889.07
7	Total Cost	39889.07
8	Utilization Cost	15566.47

This line entry represents the country Australia. There are 2 Intra-Country circuits in the backbone, but the pricing was not found (0.00). Similarly, there are 5 Inter-Country circuits with a cost of 39889.07. Adding the Inter- and Intra-Country circuit costs, the total cost is 39889.07. Utilization cost is determined by multiplying the utilization of those links by the total cost.

AVRGCOST Report

The Average Link Cost Report, AVRGCOST.x, shows summary information for site-to-site links. Information is decomposed into Cost/Kbit, A2ZCost/KB, and Z2Acost/KB. The following section shows a sample of the AVRGCOST report and an explanation.

Sample AVRGCOST Report

```
******************************************************************
*
* AVERAGE LINK COST REPORT
*                            Currency= DL(American Dollar)
*                                Date= 3/9/99 11:13
*  Number of Site= 81
*  Site= SiteN1 has 1 nodes
*       N1,
*  Site= SiteN2 has 1 nodes
*       N2,
*  Site= SiteN3 has 1 nodes
*       N3,
******************************************************************
#FromSite   ToSite          Cost/KB     A2ZCost/KB     Z2ACost/KB
SiteN1     SiteN2             4.01        0.00        0.00
SiteN1     SiteN3             3.92        0.00        0.00
SiteN1     SiteN4             2.49        0.00        0.00
SiteN1     SiteN5             4.08        0.00        0.00
```

```
SiteN1      SiteN7            10.11        0.00        0.00
SiteN1      SiteN8             4.67        0.00        0.00
SiteN1      SiteN9             3.40        0.00        0.00
SiteN1      SiteN13           13.30        0.00        0.00
SiteN1      SiteN21            2.41        0.00        0.00
SiteN1      SiteN46            7.53        0.00        0.00
SiteN1      SiteN53            5.14        0.00        0.00
SiteN1      SiteN54           62.95       34.81       28.14
```

Explanation: AVRGCOST.x Report

The AVRGCOST report presents summary Cost/KB information for each site in the network. In the first entry in the sample above:

```
SiteN1      SiteN2             4.01        0.00        0.00
```

the Cost/KB between SiteN1 and SiteN2 is 4.01. The A2Z Cost/KB and Z2A Cost/KB are both 0 to represent that the A2Z and Z2A costs were not broken down between SiteN1 and SiteN2.

In the 12th entry, the Cost/KB between SiteN1 and SiteN54 is displayed. The Cost/KB is 62.95 with 34.81 attributed to the A2Z Cost/KB and 28.14 from the Z2A Cost/KB.

```
SiteN1      SiteN54           62.95       34.81       28.14
```

LINKLOAD Report

The Link Load Report, LINKLOAD.x, lists traffic loads and trunk utilization for each link.

Sample Link Load Report

```
*********************************************************
*
*     LINK LOAD REPORT        runcode=mpls-fish
*
*     Layer 3 (Demand Layer)
*
*                 Demand Load
File=/space/wandl/sample/IP/fish/trafficload.mpls-fish
*********************************************************
    Failure Simulation not performed

TrkUtil= Demand_Load/Raw_Trk_Bw
Demutil= Demand_Load/Provisioned_Dem_BW

LINK1       ATL         HOU         ATT          1          OC3

                  A2Z     A2Z     A2Z         Z2A     Z2A     Z2A
                 Load TrkUtil Demutil        Load TrkUtil Demutil
Provision     24.082M   0.15     1.00     3.719M   0.02     1.00
WorstLoad     21.650M   0.14     0.90     3.347M   0.02     0.90
----------------------------------------------------------------------
LINK2       ATL         WDC         ATT          1          OC3

                  A2Z     A2Z     A2Z         Z2A     Z2A     Z2A
                 Load TrkUtil Demutil        Load TrkUtil Demutil
Provision    176.420M   1.13*    1.00   122.305M   0.79     1.00
WorstLoad    158.781M   1.02*    0.90   110.071M   0.71     0.90
----------------------------------------------------------------------
```

```
LINK3      BOS        DET        ATT         1         OC3

                  A2Z     A2Z     A2Z        Z2A     Z2A     Z2A
                 Load TrkUtil Demutil       Load TrkUtil Demutil
Provision     66.318M   0.43     1.00    79.417M   0.51     1.00
WorstLoad     58.758M   0.38     0.89    70.543M   0.45     0.89
---------------------------------------------------------------------
LINK4      BOS        NYC        ATT         1         OC3

                  A2Z     A2Z     A2Z        Z2A     Z2A     Z2A
                 Load TrkUtil Demutil       Load TrkUtil Demutil
Provision     35.683M   0.23     1.00    20.188M   0.13     1.00
WorstLoad     32.114M   0.21     0.90    18.168M   0.12     0.90
```

DIVPATH Report

The Primary and Backup Path Generation Report, DIVPATH.x, contains information on each demand’s primary route and its backup diversity route, as well as other detailed information pertaining to that demand. The following is a sample of a DIVPATH report and an explanation of the fields in this report.

Sample Primary and Backup Path Generation Report

```
*****************************************************
Primary And Backup Route Generation Report
*****************************************************
# Failed to find diversity route for flow1(ATL-BOS)
#flow1,,ATL,,,ATLANTA,US,,BOS,,,BOSTON,US,  730017,"R,A2Z",Don't Care,Don't Care,Don't
Care,"02,02",ATL--WDC[--CHI--DET--]BOS,4354,22,no,no,,
flow1,,ATL,,,ATLANTA,US,,BOS,,,BOSTON,US,
730017,"R,A2Z,PS(ATL-LINK2-LINK15B-LINK14-LINK4-BOS),PBK(ATL-LINK1-LINK9-LINK5-LINK7-LINK3-BOS)",Don't
Care,Don't Care,Don't Care,"02,02", # ATL--WDC--PHI--NYC--BOS,0,0,no,no,Group : Site diversity,
flow2,,ATL,,,ATLANTA,US,,CHI,,,CHICAGO,US,
730017,"R,A2Z,PS(ATL-LINK1-LINK9-LINK5-CHI),PBK(ATL-LINK2-LINK8-CHI)",Don't Care,Don't Care,Don't
Care,"02,02",ATL[--HOU--DAL--]CHI,3126,19,no,no,Group : No diversity,
flow3,,ATL,,,ATLANTA,US,,DAL,,,DALLAS,US,
418017,"R,A2Z,PS(ATL-LINK1-LINK9-DAL),PBK(ATL-LINK2-LINK8-LINK5-DAL)",Don't Care,Don't Care,Don't
Care,"02,02",ATL--HOU--DAL,1871,10,no,no,Group : No diversity,
flow4,,ATL,,,ATLANTA,US,,DEN,,,DENVER,US,
520016,"R,A2Z,PS(ATL-LINK1-LINK9-LINK5-LINK6-DEN),PBK(ATL-LINK18-LINK13-LINK16-LINK10-DEN)",Don't
Care,Don't Care,Don't Care,"02,02",ATL[--HOU--DAL--]CHI--DEN,5339,29,no,no,Group : No diversity,
# Failed to find diversity route for flow5(ATL-DET)
#flow5,,ATL,,,ATLANTA,US,,DET,,,DETROIT,US,  418017,"R,A2Z",Don't Care,Don't Care,Don't
Care,"02,02",ATL--WDC[--CHI--DET--]BOS--DET,5750,29,no,no,,
flow5,,ATL,,,ATLANTA,US,,DET,,,DETROIT,US,
418017,"R,A2Z,PS(ATL-LINK1-LINK9-LINK5-LINK7-DET),PBK(ATL-LINK2-LINK15B-LINK14-LINK4-LINK3-DET)",Don't
Care,Don't Care,Don't Care,"02,02", # ATL--HOU--DAL--CHI--DET,0,0,no,no,Group : Site diversity,
flow6,,ATL,,,ATLANTA,US,,HOU,,,HOUSTON,US,
418017,"R,A2Z,PS(ATL-LINK1-HOU),PBK(ATL-LINK2-LINK8-LINK5-LINK9-HOU)",Don't Care,Don't Care,Don't
Care,"02,02",ATL--HOU,1438,7,no,no,Group : No diversity,
flow7,,ATL,,,ATLANTA,US,,LAX,,,LOSANGELOS,US,
520016,"R,A2Z,PS(ATL-LINK18-LAX),PBK(ATL-LINK1-LINK11-LINK12-LAX)",Don't Care,Don't Care,Don't
Care,"02,02",ATL--LAX,3000,19,no,no,Group : No diversity,
```

Explanation: DIVPATH.x

The Primary and Backup Path Generation Report can be accessed when running text-mode of the Bbdsgn program. It is generated when the user selects “Primary/Backup Path Generation Report” in the Report menu of the Bbdsgn main menu. This report provides detailed information on each demand of the network and its corresponding properties such as source node, destination node, bandwidth requirement, priorities, and most importantly, its primary path route and its backup route.

DEMANDCOS Report

The Demand CoS Performance Report, DEMANDCOS.x, is a performance report for class of services of demands. The parameters of this report is selected via the Report Manager’s CoS report options window when the user first selects the CoS Demands Report. These options allow the user to select the normal or peak performance level, CoS classes to view, and the period: provisioned, all or worst.

The fields of the Demand CoS are: Demand Name, Node A, Node Z, Bandwidth, Policy Class, Direction, Load, Delay (in ms), and Drop bandwidth. Drop bandwidth is the part of the load that is dropped due to the network congestion.

Note that the Planned Bandwidth, worst delay and loss ratios data can be summarized per VPN if you have the VPN license and you select “All” for the Report Options owner.

Below is a part of a sample Demand CoS Report.

Sample Demand CoS Report

```
######################################################
#
#    Demand CoS Performance Report
#
#           Layer 3 (Demand Layer)
######################################################
#   Traffic Mode= Normal Load
#   Period= Provision
#   Policy Class= All
#     PDelay: Sum of node delay and link propagation delay for nodes and links
in the path
#     QDelay: Sum of Queuing Delay
#   ProvLoad: Bandwidth defined in Demand's bandwidth field
#   WorstLoad: Highest Bandwidth among the load in all traffic periods
#   Prov QDelay: Queuing Delay based on ProvLoad and CoS Policies
#  Worst QDelay: Highest delay value among all the periods
#
#
#OwnerName,#Demand,  ProvBW,WorstPDelay,WorstQDelay,WorstLossRatio
# VPN3,     60, 18.000M,     100.00,       25.58,    0.74,
# VPN4,      6,  1.200M,      30.00,        0.09,    0.00,
# VPN1,      6,600.000K,      84.00,        0.65,    0.00,
# VPN2,      6,600.000K,      10.00,        0.05,    0.00,
Owner,DemandName,Node,Node,BW,PolicyClass,Dir,ProDelay, Load,QDelay,DropBW,
VPN1,VPN1_1,PE2_AMB,PE1_BRB,100.000K,-,A2Z,   84,100.000K,  0.65,0,
VPN1,VPN1_2,PE1_BRB,PE2_AMB,100.000K,-,A2Z,   84,100.000K,  0.65,0,
VPN1,VPN1_3,PE2_AMB,PE1_BRB,100.000K,-,A2Z,   77,100.000K,  0.65,0,
VPN1,VPN1_4,PE1_BRB,PE2_AMB,100.000K,-,A2Z,   84,100.000K,  0.65,0,
VPN1,VPN1_5,PE2_AMB,PE1_BRB,100.000K,-,A2Z,   77,100.000K,  0.65,0,
VPN1,VPN1_6,PE1_BRB,PE2_AMB,100.000K,-,A2Z,   77,100.000K,  0.65,0,
VPN2,VPN2_7,PE1_FRA,PE2_FRA,100.000K,-,A2Z,   10,100.000K,  0.05,0,
VPN2,VPN2_8,PE2_FRA,PE1_FRA,100.000K,-,A2Z,   10,100.000K,  0.05,0,
VPN2,VPN2_9,PE1_FRA,PE2_FRA,100.000K,-,A2Z,   10,100.000K,  0.05,0,
VPN2,VPN2_10,PE2_FRA,PE1_FRA,100.000K,-,A2Z,   10,100.000K,  0.05,0,
VPN2,VPN2_11,PE1_FRA,PE2_FRA,100.000K,-,A2Z,   10,100.000K,  0.05,0,
VPN2,VPN2_12,PE2_FRA,PE1_FRA,100.000K,-,A2Z,   10,100.000K,  0.05,0,
VPN3,VPN3_19,PE1_AMB,PE1_FRB,300.000K,-,A2Z,   62,300.000K, 25.43,222.717K,
VPN3,VPN3_20,PE1_AMB,PE1_BRB,300.000K,-,A2Z,   80,300.000K,  0.65,0,
VPN3,VPN3_21,PE1_AMB,PE1_FKB,300.000K,-,A2Z,   71,300.000K,  0.65,0,
VPN3,VPN3_22,PE1_AMB,PE1_SKA,300.000K,-,A2Z,   65,300.000K,  0.23,0,
VPN3,VPN3_23,PE1_SKB,PE1_AMB,300.000K,-,A2Z,   62,300.000K, 25.43,222.717K,
VPN3,VPN3_24,PE1_SKB,PE1_BRB,300.000K,-,A2Z,   88,300.000K, 25.58,222.717K,
```

LINKCOS Report

The Link CoS Performance Report, LINKCOS.x, is a performance report for class of services of links. The parameters of this report is selected via the Report Manager’s CoS report options window when the user first selects the CoS Links Report. These options allow the user to select the normal or peak performance level, CoS classes to view, and the period: provisioned, all or worst.

The fields of the Link CoS report are: Link Name, Node, Interface, Policy Class, Load, Propagation Delay, Queuing Delay, and Drop bandwidth.

Below is a sample of a Link CoS Report.

Sample Link CoS Performance Report

```
#####################################################
#
#    Link CoS Performance Report
#
#           Layer 3 (Demand Layer)
#####################################################
#   Traffic Mode= Normal Load
#   Period= Provision
#   Policy Class= All
# PropDelay = Propagation Delay, unit=ms
# QDelay = Queuing delay, unit=ms
# DropBW = Bandwidth Dropped Rate per second
#
LinkName,Node,Interface,PolicyClass,    PropDelay,    Load,QDelay,DropBW,
LINK0,PE1_NY,fastethernet0/0,VOICE,    11,       -,     -,     -,
LINK0,PE1_NY,fastethernet0/0,BUSINESS-DATA,    11,       -,     -,     -,
LINK0,PE1_NY,fastethernet0/0,ECONOMY-DATA,    11,       -,     -,     -,
LINK0,PE1_NY,fastethernet0/0,,    11,       -,     -,     -,
LINK0,E123.45.67.0,,VOICE,    11,       -,     -,     -,
LINK0,E123.45.67.0,,BUSINESS-DATA,    11,       -,     -,     -,
LINK0,E123.45.67.0,,ECONOMY-DATA,    11,       -,     -,     -,
LINK0,E123.45.67.0,,,    11,       -,     -,     -,

LINK1,PE1_NY,fastethernet2/1,VOICE,    17,       -,     -,     -,
LINK1,PE1_NY,fastethernet2/1,BUSINESS-DATA,    17,       -,     -,     -,
LINK1,PE1_NY,fastethernet2/1,ECONOMY-DATA,    17,       -,     -,     -,
LINK1,PE1_NY,fastethernet2/1,,    17,       -,     -,     -,
LINK1,E321.45.67.89,,VOICE,    17,       -,     -,     -,
LINK1,E321.45.67.89,,BUSINESS-DATA,    17,       -,     -,     -,
LINK1,E321.45.67.89,,ECONOMY-DATA,    17,       -,     -,     -,
LINK1,E321.45.67.89,,,    17,       -,     -,     -,
```

BGPRPT Report

The BGP Report, BGPRPT.x, is an integrity check report that allows the user to verify that the network has no obvious BGP configuration errors. There are four sections in the BGP Integrity Check Report:

- BGP statistics: This section shows a) the total number of BGP speakers in the network, b) the total number of neighbors, c) the total number of policies, and d) the list of all ASes and the number of their BGP speakers.
- Neighbor AS Specification Error Check Report – This section shows any errors that ASes are not specified correctly. For example, router A declares that its neighbor, router B, is in AS 1243, but router B is actually in AS4312.
- Unbalanced BGP Neighbor Check Report – The BGP protocol requires that if a router, say A, declares that another router, say B, is its neighbor. Then router B also has to declare that router A is its neighbor. If not, then an unbalanced neighbor occurs. This section reports any unbalanced neighbors between BGP speakers within the network.
- BGP Mesh Connectivity Check Report – All IBGP speakers within an AS have to be fully meshed, unless route reflectors or confederation are used. This section shows if any AS is not fully meshed.

Sample BGP Integrity Check Report

The following is a sample of a BGP report that shows some BGP configuration errors.

```
******************************************************
*     BGP Integrity Check Report
******************************************************
-- 17 BGP speakers,89 neighbors,283 members,183 policies
-- 3 local AS:
ASno 222: 9 routers
ASno 111: 7 routers
ASno 555: 1 routers

*   *    *   *    *
Neighbor AS Specification Error Check Report

  AS   Location   Nbr_AS     Nbr_IP_Addr  Nbr-Location ValidAS Comments
  111       X39      224    69.49.226.34           Q39     222
***  1 AS specification errors

*   *    *   *    *
Unbalanced BGP Neighbor Check Report

 # Unbalanced BGP Neighbor = 2
   AS             Location   Nbr_AS       Nbr-Location
   111                  S39      111                    X39
   111                  W39      111                    X39

*   *    *   *    *

IBGP Mesh Connectivity Check Report
AS 222: passed mesh connectivity checking
AS111: S39 is not defined as X39's neighbor
AS111: W39 is not defined as X39's neighbor
 AS 111: 2 neighbor definition missing
AS 555: passed mesh connectivity checking
```

In the example above, the Neighbor AS Specification Error Check Report shows that there is an error in the node (Location) X39. The neighbor node(Nbr-Location) is Q39 and the neighbor AS (Nbr_AS) is 224, which should be 222 as shown in the ValidAS field.

The Unbalanced BGP Neighbor Check Report shows that there are two unbalanced neighbors. On the first record S39 declares that X39 is its neighbor but X39 does not declare that S39 is its neighbor. The second record shows a similar error.

The IBGP Mesh Connectivity Check Report shows that the AS111 is not fully meshed. The reason is S39 and W39 are not defined as neighbors of X39.

It is recommended that all errors reported in the BGP Report file get fixed before carrying on further analysis. One way to do it is to correct the errors on the configuration files and then run through getipconf again.

SWITCHCONN Report

The Switch Connection Statistics Report provides information on the amount of local and nonlocal traffic. The fields and their descriptions are given below:

Field	Description
NodeName	Name of the Node
#LocalDmd	Number of demands from this node to itself
LocalBW	Bandwidth of demands from this node to itself
#NonLocalDmd	Number of demands originating or destinating at this node
NonLocalBW	Bandwidth of demands originating or destinating at this node
#TransitDmd	Number of demands going through the node but not originating or destinating at it
#TransitDmd(Max)	Max number of transit demands at the switch during peak failure simulation.

MAXLKUTIL Report

The Peak Link Utilization Report, MAXLKUTIL.x, is generated when links are resized. The report consists of the maximum link utilization for each link in any single link or node failure. It lists the previous link type as well as the new link type. The new link type is of a smaller bandwidth which does not impact in performance of the network for any single line or node failure. The following is an example of the MAXLKUTIL report.

```
#########################################################################
*
*     LINK RESIZE REPORT      runcode=x
*
#########################################################################

#                            Currency= DL(American Dollar)

#    Diversity Design Parameters: sitedvpri=0, linkdvpri=0
#    Link Type Candidates: Types in tariff tables
#
#    TotalBw = (Link Capacity - Link Overhead). Unit=Kbits
#     UsedBw = Bandwidth utilized by demands. Unit=Kbits
#              Includes bandwidth required to satisfy diversity requirements
#              for single link/site failures
#    NewType = Recommended bandwidth type (if specified)
#              or required one-way bandwidth (kbits) for asymmetric trunks
#     Saving = Monthly savings if Type is replaced by NewType
#
LinkName                Vdr  Type           TotalBw    UsedBw NewType   Saving
N1-N2                   ATT  T1              1484.0    1482.1
N1-N2                   USS  T1              1484.0    1472.0
N11-N22                 ATT  T1              1484.0     722.4  FT768K      2990
N1-N4                   WTG  T1              1484.0     139.2  FT256K      1838
N1-N5                   WTG  T1              1484.0    1484.0
N1-N5                   WTG  T1              1484.0    1481.6
N1-N5                   WTG  T1              1484.0     387.2  FT512K       885
N1-N8                   WTG  T1              1484.0    1484.0
N1-N8                   WTG  T1              1484.0     692.0  FT768K      1260
N2-N3                   WTG  T1              1484.0     532.8  FT768K      1322
N3-N5                   WTG  T1              1484.0    1478.4
```

SIMRPT Report

The Interactive Simulation report, SIMRPT.x, logs the output of the interactive simulation performed using bbdsgn. Links and/or nodes may be brought down and then brought back up in any sequence. During the simulation, the program lists the network changes between simulations, and continues the simulation from the previous state. The network is reset to the original state when you select the reset environment option.

Because this file is not automatically generated, the user will need to select 'Set Options and Trace Mode' from the Simulation & Failure Analysis Menu in order to be enable the tracefile. A sample SIMRPT.x report is shown in the next section with a corresponding explanation.

Sample SIMRPT.x Report

```
Simulation Environment:
*---------------------------------------------------------------------------

*   Equipment = Generic
* Simulation Options Used:
*   Algorithm: Shortest Path Algorithm
*   Link distance = User Defined
*   randomflag = hardware default,* Path is selected randomly among paths
                     of the same length during path selection
*   Max Hop Allowed= 12,
*   Max call setup retry count = 8,
*   Path placement order= High priority demands first, scramble

   Total demand count=130, bandwidth=4184.80K
    --- Load Distribution According to Priority ---
    Pri #ckt     Bw(bit)     Bw%,   Pri #ckt     Bw(bit)     Bw%,
     10   13      2.752M   65.76%,    12  117      1.433M   34.24%,


   Max Hop: maximum hops of the demands placed
   Avg Hop: average number of hops of the demands placed
   Count: total number of demands failed to be routed
   Bandwidth: total bandwidth of demands failed to be routed
   BwRat: percentage of the total demand bandwidth failed to be routed
   HPri: highest priority of the demands failed to be routed
```

```
Simulation Environment:
  Node failure occurred at: N1(OAKLAND),
   Link failure occurred at:
     N1         N7         LEC         1          T1 : 1 down
     N1         N8         LEC         1          T1 : 1 down
     N1         N2         WTG         1          T1 : 1 down
Start Simulation:
   All 130 demands placed before configuration changes
    75 more demand(s) are disconnected due to new failures
 RN01N07        N01        N07     9.6K R 12,12  N1-N7 #DISCONNECTED
 RN01N07        N01        N07     9.6K R 12,12  N1-N7 #FAILED
 RN01N08        N01        N08     9.6K R 12,12  N1-N8 #DISCONNECTED
 RN01N08        N01        N08     9.6K R 12,12  N1-N8 #FAILED
 RN02N07        N02        N07     9.6K R 12,12  N2--N1-N7 #DISCONNECTED
 RN02N07        N02        N07     9.6K R 12,12  N2--N3--N7 #REROUTED,
 RN01N08        N01        N08    19.2K R 12,12  N1-N8 #DISCONNECTED
…
 VN01N05        N01        N05     384K V 10,10  N1-N8--N6--N5 #DISCONNECTED
 VN01N05        N01        N05     384K V 10,10  N1-N8--N6--N5 #FAILED
 VN01N08        N01        N08      64K V 10,10  N1-N8 #DISCONNECTED
 VN01N08        N01        N08      64K V 10,10  N1-N8 #FAILED

SUMMARY: Max  Avg  ..Info on Failed Demands..
         Hop  Hop Count  Bandwidth  BwRat HPr
prev:      3  1.5     0          0   0.0%  0
current:   3  1.9    65     3.271M  78.1% 12
--- Demands terminating at failed nodes: 65 (bw=3.271M)
--- Demand passing through failed nodes: 10 (bw=166.400K)
    Demand that failed to be rerouted: 0 (bw=0)
```

Explanation: SIMRPT.x Report

The environment of the simulation run is given at the beginning of the report. In the above example, the network was reset to the INITIAL STATE, and then node N1 (OAKLAND) was brought down. As a result of the failure of node N1, the links terminating/originating at N1 were also brought down. These links are:

```
Link failure occurred at:
     N1         N7         LEC         1          T1 : 1 down
     N1         N8         LEC         1          T1 : 1 down
     N1         N2         WTG         1          T1 : 1 down
```

The simulation was then performed. The statement:

```
All 130 demands placed before configuration changes
```

indicates that none of the 130 demand paths were disconnected prior to the occurrence of the new network failure. If 3 of the paths had not been placed before the simulation, the following message would have been displayed:

```
3 out of 2058 demands not placed before configuration changes
```

The statement:

```
5 more demand(s) are disconnected due to new failures
```

indicates that a total of 5 circuits were disconnected due to the node failure at N1, and resulting link failures.

The hardware has the responsibility to reroute these circuits. Disconnected circuits are sorted according to priority. The circuits with the same priority rank are rerouted in random order. bbdsgn then looks for new routes for these circuits in the sorted order. While attempting to place disconnected circuits, the program may grab bandwidth from lower priority circuits. These lower priority circuits that have been disconnected in this manner are referred as having been 'bumped'. Circuits that have been bumped are added to the list of circuits that are to be rerouted.

The original and current status of the paths being disconnected are printed in the section following the environment parameter section. Each circuit is printed twice, the original state followed by the current state. In the original state, the reason the circuit was disconnected is printed at the end of the circuit path description. Possible values include DISCONNECTED and BUMPED. If a new path is found for the circuit, then the new path is listed and the status message REROUTED is printed. If the path failed to be routed, then the old path is printed followed by the status message FAILED.

Note that circuits not affected by the network failure are not printed in this report.

A summary is given at the end of the report. The meanings of the fields are as follows:

Field	Description
Max Hops	Maximum number of hops allowed, as specified in the maxhop parameter
Avg Hops	Average number of hops for the circuits that were routed
Failed Count	Number of circuits that could not be rerouted
Failed Bandwidth	Total bandwidth of the circuits that could not be rerouted
Failed BwRat	Ratio of failed bandwidth to the total demand bandwidth
HPr	Maximum priority of the circuits that failed to be rerouted.

LKFAIL.x Report

The Link Failure Simulation Report, LKFAIL.x, is a canned failure analysis report. Every link in the network is taken down and brought back up once to see the impact of the single link failure. The network is reset to INITIAL STATE before it brings another link down/up.

When a link is taken down, all the circuits on that link are rerouted. Depending on how they are defined, they may grab bandwidth from other lower priority circuits. As a result of the single link failure, some circuits that were not originally routed on this link may be bumped due to the overflow situation caused by the link failure. When the link returns to the active state, the circuits that were bumped or disconnected may not be able to access the bandwidth that is currently available. Circuits that have been routed successfully using new paths will not be rerouted when the link is brought back to service. This is the reason links are brought back up before proceeding to the next link. In this report, the network is reset to the original state only before taking another link down.

A sample LKFAIL.x report is shown below with a corresponding explanation.

Sample LKFAIL.x Report

```
*************************************************************************
*         LINK FAILURE SIMULATION REPORT -- run pl,    3/12/99 13:58
*************************************************************************

* Simulation sequence:
*       1. Reset the network to initial state
*       2. Bring all links between a node pair down, simulate
*       3. Bring all links between a node pair up, simulate
*       4. Go to step 1



*--------------------------------------------------------------------------

*   Equipment = Generic
* Simulation Options Used:
*   Algorithm: Shortest Path Algorithm
*   Link distance = User Defined
*   randomflag = hardware default,* Path is selected randomly among paths
                     of the same length during path selection
*   Max Hop Allowed= 12,
*   Max call setup retry count = 8,
*   Path placement order= High priority demands first, scramble

   Total demand count=130, bandwidth=4184.80K
    --- Load Distribution According to Priority ---
    Pri #ckt     Bw(bit)     Bw%,   Pri #ckt     Bw(bit)     Bw%,
     10   13     2.752M   65.76%,    12  117     1.433M   34.24%,



   Max Hop: maximum hops of the demands placed
   Avg Hop: average number of hops of the demands placed
   Count: total number of demands failed to be routed
   Ratio: ratio of failed demands vs total number of demands
   Bandwidth: total bandwidth of demands failed to be routed
   Bw-Ratio: percentage of the total demand bandwidth failed to be routed
   MaxPri: highest priority of the demands failed to be routed

   ^ : bandwidth available, but demand failed to be routed by hardware

--------------------------------------------------------------------------

 HPr:  Highest priority of demands not placed
 BwRat: (Total bandwidth of demands not placed)/Total bandwidth

                        Count        Max  Avg  ..Info on Failed Demands..
Link                        #   Type Hop  Hop Count  Bandwidth  BwRat HPr
N1         N2         down  1          4  1.6    11   550.400K  13.2% 12
N1         N2           up  1          4  1.6     0          0   0.0%  0
N1         N6         down  1          4  1.3    24   998.400K  23.9% 12
N1         N6           up  1          4  1.6     0          0   0.0%  0
N1         N7         down  1          4  2.0     5     1.280M  30.6% 10
N1         N7           up  1          4  2.0     0          0   0.0%  0
N1         N8         down  1          3  1.6     5     1.280M  30.6% 10
N1         N8           up  1          3  1.6     2   768.000K  18.4% 10
N3         N4         down  1          4  1.6    14   508.800K  12.2% 12
N3         N4           up  1          4  1.6     0          0   0.0%  0
N3         N7         down  1          4  1.6    26   993.600K  23.7% 12
N3         N7           up  1          4  1.6     0          0   0.0%  0
N5         N6         down  1          4  1.4    12   484.800K  11.6% 12
N5         N6           up  1          4  1.6     0          0   0.0%  0
N7         N8         down  1          3  1.5     5   960.000K  22.9% 10
N7         N8           up  1          4  1.5     0          0   0.0%  0

--- Worst case occurred link between N1 and N7.
    Demands that could not be routed:    5 (bw=1.280M)
--- Highest priority of demands not routed: 12
```

Explanation: LKFAIL.x Report

The simulation parameters are listed at the beginning of the report. In the above example, the shortest path algorithm is used and the maximum number of hops allowed in a path is 12. There are 130 paths in the backbone. Total bandwidth of the circuits is 4,184.8 Kb. Of the 130 circuits, thirteen are priority 10. Aggregate bandwidth for these circuits is 2,752 Kb, which is 65.76% of the total circuit bandwidth. The remaining 117 of 130 circuits are priority 12. Their aggregate bandwidth is 1,433 Kb, which is 34.24% of the total circuit bandwidth.

Summary information for each link failure is given following the simulation parameters section of the report. The network is reset prior to bring each link down. Note that the network reset is not explicitly indicated in the report. This report option contains summary information only. Alternate report options may be selected to generate more detailed failure analysis reports.

Descriptions of links brought down/up are listed in the following format:

```
N1          N2          down  7           9  1.7      5   160.000K   0.4%  7
N1          N2            up  7           9  1.7      0          0   0.0%  0
N1          N3          down  5           9  1.7     22   704.000K   1.6%  7
N1          N3            up  5           9  1.7      0          0   0.0%  0
N1          N4          down  1           8  1.7      0          0   0.0%  0
N1          N4            up  1           8  1.7      0          0   0.0%  0
```

In the first line entry shown above, there are 7 links between N1 and N2. All 7 of them are taken down simultaneously. The next line shows all 7 link brought back to service.

```
N1          N2            up  7           9  1.7      0          0   0.0%  0
```

The meanings of the fields in this report are the same as those described in the SIMRPT section.

Before executing the failure analysis run, the program will prompt the user to specify the level of detail for the report to be generated. Output reports may range from just summary to very detailed individual circuit information.

LINEFAIL.x Report

This is the Single Link Failure Simulation Report, LINEFAIL.x. A link is a group of private line circuits, with the same endpoints, that are leased from the same vendor and have the same bandwidth type. In the LKFAIL.x report, all the private line circuits in a link are taken down simultaneously. In the LINEFAIL.x report, only one private line is taken down for each failure simulation. The output below shows a sample from a LINEFAIL.x report.

```
N13-N71.1               down   8  1.7     0          0   0.0%  0
N13-N71.2               down   8  1.7     1    32.000K   0.1%  7
```

In this example, there are two private lines between N13 and N71. The first entry represents the failure simulation where only 1 of the 2 private lines is brought down (N71.1). The second entry represents the failure simulation where the second private line (N71.2) is brought down as well.

NDFAIL.x Report

To generate the Node Failure Simulation Report, NDFAIL.x, each node is brought down to determine the impact of node failure. The network is reset to the original state prior to taking another node down. As expected, when a node is brought down the circuits originating/terminating at that node will be disconnected. One objective of the node failure simulation is to find out whether the circuits passing through this node can be rerouted in a failure scenario, and to determine if any of the circuits that were bumped as a result failed to be rerouted.

All the demands not routed during node failure are grouped into three categories:

- **passthru**: demands that passed through the failed node or link before the failure simulation
- **bump**: demands that did not pass through the failed node or link but are preempted by other demands during the failure simulation
- **other**: demands that are not routed before and after the simulation

Sample NDFAIL.x Report

```
*************************************************************************
*       NODE FAILURE SIMULATION REPORT -- run pl,  3/12/99 15:38
*************************************************************************

  Simulation sequence:
        1. Reset the network to initial state
        2. Bring one node down and simulate
        3. Go to step 1


*---------------------------------------------------------------------------

*   Equipment = Generic
* Simulation Options Used:
*   Algorithm: Shortest Path Algorithm
*   Link distance = User Defined
*   randomflag = 1, * Path is selected randomly among paths
                     of the same length during path selection
*   Max Hop Allowed= 12,
*   Max call setup retry count = 8,
*   Path placement order= High priority demands first, scramble

   Total demand count=1464, bandwidth=42854.40K
    --- Load Distribution According to Priority ---
    Pri #ckt     Bw(bit)     Bw%,   Pri  #ckt    Bw(bit)     Bw%,
      2   10    96.000K    0.22%,     7  827    26.658M   62.21%,
      8  499     7.226M   16.86%,     9    3   168.000K    0.39%,
     10  125     8.707M   20.32%,


   Max Hop: maximum hops of the demands placed
   Avg Hop: average number of hops of the demands placed
   Count: total number of demands failed to be routed
   Ratio: ratio of failed demands vs total number of demands
   Bandwidth: total bandwidth of demands failed to be routed
   Bw-Ratio: percentage of the total demand bandwidth failed to be routed
   MaxPri: highest priority of the demands failed to be routed
   (S): links at SINGLE_END locations
   #passthru: number of pass through demands that can't be rerouted
   #bumped: number of bumped demands that can't be rerouted
   #other: number of demands not routed before and after the simulation

--------------------------------------------------------------------------

 HPr:  Highest priority of demands not placed
 BwRat: (Total bandwidth of demands not placed)/Total bandwidth

                   Max  Avg  ..Info on Failed Demands..
NODE               Hop  Hop Count  Bandwidth  BwRat HPr
N1                  10  1.8   196     6.659M  15.5% 10
    (Passthru: 19, High Priority: 8, Bumped: 29, High Priority: 7)
N2                   8  1.7   143     3.400M   7.9% 10
    (Passthru: 14, High Priority: 7, Bumped: 5, High Priority: 7)
N3                   8  1.7   143     4.339M  10.1%  8
N4                   8  1.7   122     3.203M   7.5% 10
    (Passthru: 34, High Priority: 7,)
N5                   8  1.6    86     1.145M   2.7% 10
N6                   8  1.7   113     3.179M   7.4% 10
N7                   8  1.7   161     5.045M  11.8% 10
N8                   9  1.8   227     6.879M  16.1% 10
    (Passthru: 35, High Priority: 8, Bumped: 15, High Priority: 7)
N9                   8  1.7   278     6.806M  15.9% 10
N10                  8  1.7   123     3.817M   8.9% 10
N11                  8  1.7   111     3.409M   8.0% 10
    (Passthru: 8, High Priority: 8, Bumped: 2, High Priority: 7)
N12                  8  1.6   118     3.389M   7.9% 10
    (Passthru: 14, High Priority: 7, Bumped: 5, High Priority: 7)
```

Explanation: NDFAIL.x Report

The NDFAIL.x report is similar to the LKFAIL.x report. Simulation environment parameters are listed at the beginning of the report, followed by individual node failure statistics.

In the above example,

```
N1                  10  1.8   196     6.659M  15.5% 10
    (Passthru: 19, High Priority: 8, Bumped: 29, High Priority: 7)
```

means that 196 circuits failed to be routed. The highest priority of these circuits is 10, and the aggregate bandwidth of these 196 circuits is 6.659 Mb. Of the 196 unplaced circuits, 19 were passthrough with the highest priority being 8. As well, 29 of the 196 were circuits that had been bumped and had a high priority of 7.

If the Failure Report Options had been set to option 3, Detailed Circuit Information of Failed Paths, the entry for node failure N1 would be as follows:

```
N1                  12  1.9   196     6.654M  15.5% 10
    (Passthru: 18, High Priority: 8, Bumped: 30, High Priority: 7)
 ** ckt276   (psthru),ckt277   (psthru),ckt278   (psthru),ckt279   (psthru),
 ** ckt280   (psthru),ckt281   (psthru),ckt282   (psthru),ckt283   (psthru),
 ** ckt284   (psthru),ckt285   (psthru),ckt274   (psthru),ckt286   (psthru),
 ** ckt275   (psthru),ckt289   (bumped),ckt290   (bumped),ckt291   (bumped),
 ** ckt292   (bumped),ckt293   (bumped),ckt294   (bumped),ckt288   (bumped),
 ** ckt295   (bumped),ckt287   (bumped),ckt296   (bumped),ckt297   (bumped),
 ** ckt298   (bumped),ckt299   (bumped),ckt300   (bumped),ckt328   (bumped),
 ** ckt329   (bumped),ckt355   (bumped),ckt442   (psthru),ckt448   (psthru),
 ** ckt465   (psthru),^ckt708  (bumped),ckt751   (bumped),ckt746   (bumped),
 ** ckt754   (bumped),ckt755   (bumped),ckt744   (bumped),ckt747   (bumped),
 ** ckt749   (bumped),ckt828   (bumped),ckt827   (bumped),ckt1044  (bumped),
 ** ckt1040  (bumped),ckt1041  (bumped),ckt1373  (psthru),ckt1715  (psthru),
```

This output displays the ID of the circuits that failed to be routed. The user will note that the summary statistics vary slightly from the previous example for the node failure at N1. This is related to the random order of rerouting for circuits with the same priority, in addition to issues of bin packing.

FACFAIL.x Report

The Facility Failure Simulation Report, FACFAIL.x, is the result of a facility failure simulation.

An example of the FACFAIL report is shown below:

```
*************************************************************************
*      FACILITY FAILURE SIMULATION REPORT -- run x,    8/7/96 14:16
*************************************************************************
* Simulation sequence:
*       1. Reset the network to initial state
*       2. Bring one facility down and simulate
*       3. Go to step 1
*---------------------------------------------------------------------------
*   Equipment = CASCADE
* Simulation Options Used:
*   Algorithm: Shortest Path Algorithm.
*   Link distance = Hardware Specific
*   Max Hop Allowed= 8,
*   Path placement order= High priority demands first,scramble

   Total demand count=130, bandwidth=4184.80K
     Load distribution according to priorities
    Pri #ckt   BW(Kbit)  BW%  ,   Pri #ckt   BW(Kbit)  BW%   ,
     10   13    2752.0K 65.76%,    12  117    1432.8K 34.24%,

   Max Hop: maximum hops of the demands placed
   Avg Hop: average number of hops of the demands placed
   Count: total number of demands failed to be routed
   Ratio: ratio of failed demands vs total number of demands
   Bandwidth: total bandwidth of demands failed to be routed
   Bw-Ratio: percentage of the total demand bandwidth failed to be routed
   MaxPri: highest priority of the demands failed to be routed
--------------------------------------------------------------------------
*                            Max  Avg  ..Info on Failed Demands.. #LkBw
*FacName                     Hop  Hop Count  Bandwidth  BwRat HPr OvSub
FAC1                    down   3  1.8    75    3436800  82.1% 12  0
FAC2                    down   3  1.4    50    1992000  47.6% 12  0
FAC3                    down   5  3.0   125    2264800  54.1% 12  2

** Worst case occurred at FAC1
     75 paths(bw=3436800) can't be routed
** Highest priority of paths not routed=12
** Worst link bandwidth violation occurred at FAC3
  Violation occurred at 2 links
```

DAILYFAIL.x Report

This is the Daily Random Failure Simulation Report, DAILYFAIL.x. The simulations mentioned in previous sections only perform single link failure, single line failure, and single node failure. The daily random failure simulation may be used to simulate multiple failures over several days. Failure sequences generated by the program are saved in the DAILYSEQ.x file. It can be modified and used in future simulation runs.

Sample DAILYFAIL.x Report

```
*************************************************************************
*     RANDOM DAILY FAILURE SIMULATION REPORT  -- run xx,  3/12/99 16:07
*
*     Number of Days to Simulate= 45 days
*     Number of Failure Per Day= 3
*     Failure Type = Single Line Failure
*************************************************************************

---------------------------------------------------------------------------



*--------------------------------------------------------------------------


*   Equipment = Generic
* Simulation Options Used:
*   Algorithm: Shortest Path Algorithm
*   Link distance = User Defined
*   randomflag = 1, * Path is selected randomly among paths
                     of the same length during path selection
*   Max Hop Allowed= 12,
*   Max call setup retry count = 8,
*   Path placement order= High priority demands first, scramble

   Total demand count=1464, bandwidth=42854.40K
    --- Load Distribution According to Priority ---
    Pri #ckt     Bw(bit)     Bw%,   Pri #ckt     Bw(bit)     Bw%,
      2   10    96.000K    0.22%,     7  827    26.658M   62.21%,
      8  499     7.226M   16.86%,     9    3   168.000K    0.39%,
     10  125     8.707M   20.32%,


   Max Hop: maximum hops of the demands placed
   Avg Hop: average number of hops of the demands placed
   Count: total number of demands failed to be routed
   Ratio: ratio of failed demands vs total number of demands
   Bandwidth: total bandwidth of demands failed to be routed
   Bw-Ratio: percentage of the total demand bandwidth failed to be routed
   MaxPri: highest priority of the demands failed to be routed

   (S): links at SINGLE_END locations

------------------------------------------------------------------------


 HPr:  Highest priority of demands not placed
 BwRat: (Total bandwidth of demands not placed)/Total bandwidth

                             Count         Max  Avg  ..Info on Failed Demands..
Link/Node/Node_Pair              #   Type  Hop  Hop Count  Bandwidth  BwRat HPr
N12        N71         ATT down  1 FT512K    8  1.7     0          0   0.0%  0
```

```
N3         N19         ATT down  1     T1     8  1.7    25   800.000K   1.9%  7
N20        N30         WTG down  1     T1     8  1.7    38     1.216M   2.8%  7
** RESET NETWORK TO ORIGINAL STATE     ... day 1
N1         N21         WTG down  1     T1     8  1.7     0          0   0.0%  0
N52        N56         ATT down  1 FT384K     8  1.7     0          0   0.0%  0
N4         N41         USS down  1 FT512K     8  1.7     2    64.000K   0.1%  7
** RESET NETWORK TO ORIGINAL STATE     ... day 2
N11        N36         WTG down  1     T1     8  1.7    20   664.000K   1.5%  7
N75        N78         ATT down  1 FT512K     8  1.7    20   664.000K   1.5%  7
N68        N192        ATT down  1 FT256K     8  1.7    20   664.000K   1.5%  7
** RESET NETWORK TO ORIGINAL STATE     ... day 3
N8         N13         WTG down  1     T1     7  1.7    20   664.000K   1.5%  7
N72        N79         ATT down  1 FT512K     7  1.7    20   664.000K   1.5%  7
N72        N79         ATT down  1 FT512K     7  1.7    20   664.000K   1.5%  7
** RESET NETWORK TO ORIGINAL STATE     ... day 4
N8         N43         WTG down  1 FT512K     8  1.7     0          0   0.0%  0
N190       N191        ATT down  1 FT384K     8  1.7     0          0   0.0%  0
N22        N37         USS down  1     T1     8  1.7    16   489.600K   1.1%  7
** RESET NETWORK TO ORIGINAL STATE     ... day 5
N76        N79         ATT down  1 FT512K     8  1.7     0          0   0.0%  0
N13        N30         USS down  1     T1     8  1.7    25   800.000K   1.9%  7
N11        N22         ATT down  1 FT768K     8  1.7    33     1.034M   2.4%  7
…
** RESET NETWORK TO ORIGINAL STATE     ... day 45

--- Worst case occurred day 17
    Demands that could not be routed:   64 (bw=2.240M)
--- Highest priority of demands not routed: 10
```

Explanation: DAILYFAIL.x

In the above example of Dailyfail.x, the program is instructed to generate 3 single line failures each day for 45 days. In the first day, the following links were brought down: N12-N71, N3-N19, and N20-N30. When the link N12-N71 was brought down, all circuits were able to be rerouted. When the link N3-N19 was brought down, 25 circuits failed to be rerouted. Finally, when the link N20-N30 was brought down, a total of 38 circuits failed to be rerouted.

DAILYSEQ.x Report

This is the Daily Random Failure Sequence Report, DAILYSEQ.x, that is generated after running a Random Daily Failure simulation. This file can be modified and used in future simulation runs. Examples of daily random failure sequence files are shown below.

Sample 1 - Multiple Vendor Failure

```
N8 USS down
N6 USS down
reset
N2 USS down
N3 USS down
reset
reset
N5 USS down
N7 USS down
reset
N4 USS down
N3 USS down
```

```
reset
```

Sample 2 - Multiple Link Failure

```
reset
N7 N8 LEC down 1 T1
N3 N4 USS down 1 T1
reset
N9 N7 USS down 1 T1
N1 N8 LEC down 1 T1
reset
N4 N5 USS down 1 T1
N5 N6 USS down 1 T1
reset
N2 N3 MCI down 1 T1
N5 N9 USS down 1 T1
reset
N3 N7 MCI down 1 T1
N6 N8 USS down 1 T1
Reset
```

DAILYSEQ.x Format

The user may define a custom network failure sequence and use the 'Replay Up-Down Sequence' option in the Failure Analysis Menu to simulate it.

Nodes, sites, links and vendors may be brought down/up in any sequence. To reset the network and start another sequence of simulations, the qualifier reset should be used. The following sections illustrate how a sequence file may be generated.

Node Failure

To bring a node down/up, include a line consisting of the node ID or name followed by the keyword down or up. In the following example, node 5 is first brought down, then it is brought back up.

```
N05    down
N05    up
```

When a node is brought down, all the links originating/terminating at it are automatically disconnected. Conversely, when a node is brought back to service, all the links originating/terminating at it are reconnected.

Site Failure

To bring a site down/up, include a line containing the site name followed by the keyword down or up. In the following example, site NYSITE is first brought down, then it is brought back to service.

```
NYSITE    down
NYSITE    up
```

When a site is brought down, all links originating/terminating at it are automatically disconnected. When a site is brought back to service, all links originating/terminating at it are reconnected.

Vendor Failure

A vendor failure may be specified at either a node or a site. Each line entry should consist of the node or site name, vendor name, and the keyword down or up.

Example vendor failure entries are shown below:

```
NYSITE    ATT    down
NYSITE    ATT    up
N5    ATT    down
N5    ATT    up
```

In the first line entry, all AT&T links originating/terminating at NYSITE are brought down. Similarly, in the third line entry, all AT&T links at node N5 are brought down.

Link Failure

To define a link failure in a sequence file a line entry should contain at least three fields: origination node, destination node, and the keyword up or down. Site names cannot be used to define link states. The program brings all links between the from and to nodes down/up, accordingly. If vendor names are specified, then only links leased from that vendor are brought down/up. If quantity and/or bandwidth type are defined, then only the specified number of links for that bandwidth type are brought down/up. The following formats may be used to define link failures:

```
linkname     up/down
from_node    to_node    up/down
from_node    to_node    up/down    quantity
from_node    to_node    up/down    quantity    bandwidth_type
from_node    to_node    vendor     up/down
from_node    to_node    vendor     up/down     quantity
from_node    to_node    vendor     up/down     quantity         bandwidth_type
```

UPDOWN.x Report

This report lists the demands or tunnels that were brought down as a result of a failure simulation. In situations where an upper layer network contains links that are derived from the demands of a lower layer network, this report can be generated in a lower layer network failure simulation and read in as a custom link failure simulation script in the upper layer network. Below is an example of an UPDOWN.x report:

```
##  Report Date= 2/18/2004 10:25   Runcode=mpls-fish  User=wandl
# NYC down
Voip3           down
Http1           down
Ftp97           down
RESET
# PARIS down
Intranet1       down
Dns3            down
Smpt54          down
RESET
```

When read in as a custom failure simulation script in an upper layer network, the links Voip3, Http1, and Ftp97 will be brought down first. Upon reaching the RESET statement, the program will bring back up all links, then proceed to bring down the links Intranet1, Dns3, and Smpt54.

RNDLKUTL.x Report

The Random Link Utilization Report, RNDLKUTL.x, is automatically generated after replaying an up-down sequence for failure analysis. RNDLKUTL records the maximum link utilization which occurred during the up-down sequence. The following is an example of a RNDLKUTL report.

```
#########################################################################
*
*     LINK UTILIZATION REPORT      runcode=496
*
#########################################################################
#    AvailBw = available bandwidth in the link
#    UsedBw = bandwidth used by circuit demands
#    Ovhd = link overhead
#    TotalBw = AvailBw + UsedBw + Ovhd
#    Unit = Kbits
#
Linkname                 Type         TotalBw    AvailBw     UsedBw    Ovhd
N1        -N4                   T1      1544.0     1428.8      115.2     0.0
N1        -N5                   T1      1544.0        0.0     1544.0     0.0
N1        -N5                   T1      1544.0      144.8     1399.2     0.0
N1        -N5                   T1      1544.0     1544.0        0.0     0.0
N2        -N3                   T1      1544.0     1035.2      508.8     0.0
N3        -N5                   T1      1544.0        3.2     1540.8     0.0
N3        -N5                   T1      1544.0      626.4      917.6     0.0
N3        -N7                   T1      1544.0      464.0     1080.0     0.0
N3        -N8                   T1      1544.0      856.8      687.2     0.0
N6        -N9                   T1      1544.0     1336.8      207.2     0.0
N6        -N10                  T1      1544.0        8.0     1536.0     0.0
N6        -N10                  T1      1544.0     1288.0      256.0     0.0
N6        -N10                  T1      1544.0        8.0     1536.0     0.0
```

RNDPATH.x Report

The Random Path Placement Report, RNDPATH.x, is automatically generated after replaying an up-down sequence for failure analysis. RNDPATH contains the path placement information for each demand at the end of the up-down sequence. As links and nodes are brought down/up, the path placements will obviously change due to rerouting. A sample RNDPATH report is shown below:

```
1000023A   N1         N2           192000 R 01,01 N1--N5--N3--N2
1000023A   N1         N2           192000 R 01,01 N1--N5--N3--N2
1000023B   N1         N2            19200 R 01,01 N1--N5--N3--N2
1000023C   N1         N2             9600 R 01,01 N1--N5--N3--N2
1000023C   N1         N2             9600 R 01,01 N1--N5--N3--N2
1000023C   N1         N2             9600 R 01,01 N1--N5--N3--N2
1000023C   N1         N2             9600 R 01,01 N1--N5--N3--N2
1000023C   N1         N2             9600 R 01,01 N1--N5--N3--N2
10000157   N1         N2             9600 R 00,00 N1--N5--N3--N2
10000174   N1         N2             9600 R 00,00 N1--N5--N3--N2
10000061   N1         N3            19200 R 01,01 N1--N5--N3
10000062   N1         N3            19200 R 01,01 N1--N5--N3
```

SIMPLACE.x Report

This is the Path Placement Simulation Report, SIMPLACE.x. Circuits can be placed even without being given explicit path information. The program will attempt to place the circuits in a specified number of iterations, as reflected by the max call setup retry count. This information is useful in deciding where extra private lines should be added, and when the path-select feature is not supported by the hardware.

A sample SIMPLACE report is shown below.

Sample SIMPLACE.x Report

```
*************************************************************************
*      PATH PLACEMENT SIMULATION REPORT   -- run xxx,  date 3/12/99 17:34
*************************************************************************
* Simulation Options Used:
*   Algorithm: Shortest Path Algorithm
*   Link distance = User Defined
*   randomflag = hardware default,* Path is selected randomly among paths
                     of the same length during path selection
*   Max Hop Allowed= 12,
*   Max call setup retry count = 8,
*   Path placement order= High priority demands first, scramble

   ^ : bandwidth available, but demand failed to be routed by hardware

--------------------------------------------------------------------------

   HPr:  Highest priority of demands not placed
   BwRat: (Total bandwidth of demands not placed)/Total bandwidth
               Max  Avg  ..Info on Failed Demands..
Iteration      Hop  Hop Count  Bandwidth  BwRat HPr
iteration   1:   3  1.5     0          0   0.0%  0
iteration   2:   3  1.5     0          0   0.0%  0
iteration   3:   3  1.5     0          0   0.0%  0
iteration   4:   3  1.5     0          0   0.0%  0
iteration   5:   7  1.5     0          0   0.0%  0
```

In the sample report shown above, the program simulated placing all the circuits over 5 iterations. For each of the iterations, 0 of the demands failed to be routed.

TRAFFICLOAD.x Report

The Link Load Report, trafficload.x, provides summary information for each link in the network based on defined traffic PVC loads as specified by the user in the traffic load file, or based on default program settings. Information provided for each link include the amount of bandwidth planned as well as the worst case bandwidth utilization, for each direction of the link. If a Peak Simulation is run on the network, additional fields in the Link Load report will include the peak bandwidth provisioned, as well as the worst case peak bandwidth. As the user will recall, the Peak Simulation determines the maximum bandwidth that may be routed over a link in any worst case failure scenario.

A sample Link Load Report file is shown below.

```
*********************************************************
*     LINK LOAD REPORT        runcode=1098
*
*                 PVC Load File=trafficload
```

```
********************************************************
Failure Simulations Performed:
     1.  Automatic Single Link Failure Simulation
TrkUtil= Load/Raw_Trk_Bw
PVCutil= Load/(PVC BW calculated from path specification)
LK624      N12801     N12802     DEF         1        T1 K=0.50
                 A2Z     A2Z     A2Z        Z2A     Z2A     Z2A
                 Load TrkUtil PVCutil      Load TrkUtil PVCutil
Provision      1.018M   0.70    0.50     1.018M   0.70     0.50
WorstLoad    188.152K   0.13    0.09   116.184K   0.08     0.06
Peak_Prov      1.446M   0.99*   0.50     1.446M   0.99*    0.50
WorstPeakLD  272.374K   0.19    0.09   522.098K   0.36     0.18

---------------------------------------------------------------------
LK4036     N12804     N12806     DEF       1  HSSI15.79M K=0.50
                 A2Z     A2Z     A2Z        Z2A     Z2A     Z2A
                Load TrkUtil PVCutil       Load TrkUtil PVCutil
Provision    12.477M   0.83    0.50     12.477M   0.83     0.50
WorstLoad     2.886M   0.19    0.12     15.386M   1.03*    0.62
Peak_Prov    14.519M   0.97*   0.50     14.519M   0.97*    0.50
WorstPeakLD   5.367M   0.36    0.18     16.579M   1.11*    0.57
```

The CSV format for the Traffic Load Report will print the link load for all the different periods when the report format is set to CSV. The report format may be set from the Report Options Menu. Otherwise, only the planned and worst load information is included in the Traffic Load Report.

For more information on the usage of this feature, please consult the Traffic Load Feature Manual (Part No: TFL8).

PATHDELAY.x Report

The Path Delay Information Report, PATHDELAY.x, can be generated in the JAVA GUI by selecting the Generate Peak Utilization Report option in the Scripts menu (Simulation>Scripts) along with the script of your choice.

In bbdsgn, it can be generated by the following menu options from the main menu: 8. Failure Simulation, 4. Peak Utilization & Load Analysis, 5. Peak Utilization Report.

Here is a portion of a sample max path delay report:

```
**************************************************
*     Path delay Information Report
**************************************************
Failure Simulations Performed:
*    1.  Automatic Link Failure Simulation
*  PropDelay = propagation delay of the path at normal situation
*  Failcnt = # of times this path was disconnected during failure simulation
*  WorstDelay= Worst propagation delay of the alternate routes occurred during failure simulation
*              0 means paths not rerrouted during failure simulation
*  Simeventype,SimEvent= Occasion where worst delay occurred
Pathname,From,To,Bandwidth,Type,Priority,Path,AdmCost,PropDelay,Failcnt,WorstDelay,SimEventType,SimEvent
flow1,ATL,BOS,730.017K,"R,A2Z","02,02",ATL--WDC--PHI--NYC--BOS,1442,13,0,30,LKFAIL,ATL-WDC,
flow2,ATL,CHI,730.017K,"R,A2Z","02,02",ATL--WDC--CHI,2481,13,0,20,LKFAIL,ATL-WDC,
flow3,ATL,DAL,418.017K,"R,A2Z","02,02",ATL--HOU--DAL,1871,11,0,22,LKFAIL,ATL-HOU,
flow4,ATL,DEN,520.016K,"R,A2Z","02,02",ATL--WDC--CHI--DEN,4694,23,0,36,LKFAIL,CHI-DEN,
flow5,ATL,DET,418.017K,"R,A2Z","02,02",ATL--WDC--PHI--NYC--BOS--DET,2838,20,0,23,LKFAIL,ATL-WDC,
flow6,ATL,HOU,418.017K,"R,A2Z","02,02",ATL--HOU,1438,8,0,25,LKFAIL,ATL-HOU,
...
```

For each demand, there is a report of the propagation delay of the path in the normal situation, the worst propagation delay, and the number of times the demand was disconnected during the failure simulation. Then the particular occasion where the worst propagation delay occurred is given.

For example, in the below line, in all the link failures, the normal propagation delay was 13 milliseconds and the demand was never disconnected. The worst propagation delay was 30 milliseconds and it occurred when simulating the failure of link ATL-WDC.

```
flow1,ATL,BOS,730.017K,"R,A2Z","02,02",ATL--WDC--PHI--NYC--BOS,1442,13,0,30,LKFAIL,ATL-WDC,
```

PeakSimSummary.x Report

The Peak Simulation Summary Report can be generated by running failure simulation from menu Simulation > Predefined Scenarios. This report provides a summary of failure events and the impact the event has on demands, tunnels, and links.

Here is a portion of a sample peak simulation summary report:

```
***********************************************
*  PEAK SIMULATION STAT SUMMARY
***********************************************
##  Software Release= 6.0.1, 64 bits,  Compilation Date= 20120418
##  Platform=i86pc, OS=SunOS 5.10
##  Report Date= 4/20/2012 10:45   Runcode=mpls-fish  User=wandl
#
# Up/Down:  Up or Down operation
# Layer:  Demand or Tunnel
# #Impact:  Demand or Tunnel impacted
# ImpactBW:  Total bandwidth of demand or tunnel impacted
# MaxHop:  Max path hop count after failure
# AvgHop:  Average path hop count after failure
# #failed:  Number of disconnected flows (Demand or tunnel terminated at failed
nodes not included)
# FailedBW:  Total bandwidth of disconnected flows
# FailedBwPct:  FailedBW/TotalFlowBandwidth percentage
# Hpr:  Highest priority of failed flows
# #LkBwOvSub:  #links where bandiwdth oversubscription occurred
#      i.e. usedBW> (1-fatpct)*trunkBandwidth where fatpct= 0.00%
# #Terminated: # flows terminated at failed node
# TermBW: Total bandwidth of flows terminated at failed node
#
SimType,SimEvent,UP/Down,Layer,#Impact,ImpactBW((Mbit)),#failed,FailedBW((Mbit))
,FailedBwPct,HPr,#LkBwOvSub,MaxHop,AvgHop,#Terminated,TermBW
NDFAIL,ATL,down,tunnel,382,3218.38,0,0.0,0.0,0,0,5,3.5,1,1.0,
NDFAIL,ATL,down,demand,382,3218.38,0,0.0,0.0,0,5,8,3.2,149,1066.279,
NDFAIL,BOS,down,tunnel,201,1862.274,0,0.0,0.0,0,0,5,3.7,2,25,
NDFAIL,BOS,down,demand,201,1862.274,0,0.0,0.0,0,4,6,2.8,156,1729.18,
NDFAIL,CHI,down,tunnel,456,5473.657,3,30,83.3,2,0,0,0.0,2,6.0,
NDFAIL,CHI,down,demand,456,5473.657,0,0.0,0.0,0,4,9,3.6,156,3994.123,
NDFAIL,DAL,down,tunnel,258,1774.568,3,11,30.6,2,0,3,3.0,0,0.0,
NDFAIL,DAL,down,demand,258,1774.568,0,0.0,0.0,0,6,7,3.0,138,744.512,
NDFAIL,DEN,down,tunnel,183,1590.976,0,0.0,0.0,0,0,5,3.4,0,0.0,
NDFAIL,DEN,down,demand,183,1590.976,0,0.0,0.0,0,5,7,3.2,148,1066.597,
```

PeakSimLink.x Report

The Peak Simulation Link Report can be generated by running failure simulation from menu Simulation > Predefined Scenarios. This report provides information on links that are oversubscribed from the failure simulation.

Here is a portion of a sample peak simulation link report:

```
************************************************
*  LINK OVERSUBSCRIPTION STAT SUMMARY
************************************************
##  Software Release= 6.0.1, 64 bits,  Compilation Date= 20120418
##  Platform=i86pc, OS=SunOS 5.10
##  Report Date= 4/20/2012 10:45   Runcode=mpls-fish  User=wandl
# LinkBW:  Link bandwidth
# UsedBw:  Bandwidth allocated for Demands after failure
# UtilPct:  UsedBW/LkBW * 100
# WorstLoad:  Max traffic load sum of Demands routed over the link after failure
# WorstLoadUtilpct:   WorstLoad/LkBW * 100
# Note: only links with UtilPct or WortLoadUtilPct exceeds (1-fatpct) is 
displayed
#       fatpct= 0.00%
#
SimType,SimEvent,Layer,LinkName,NodeA,InterfaceA,NodeZ,LinkBW(Mbit),UsedBw(Mbit)
,UtilPct,WorstLoad((Mbit)),WorstLoadUtilPct
NDFAIL,ATL,Demand,LINK5,DAL,,CHI,2488.32,2955.666,118.78,168.963,6.79,
NDFAIL,ATL,Demand,LINK8,CHI,,WDC,2488.32,3326.187,133.67,180.396,7.25,
NDFAIL,ATL,Demand,LINK9,HOU,,DAL,2488.32,2677.466,107.60,151.626,6.09,
NDFAIL,ATL,Demand,LINK13,SJC,,LAX,2488.32,2913.956,117.11,105.575,4.24,
NDFAIL,ATL,Demand,LINK14,PHI,,NYC,2488.32,2788.812,112.08,116.277,4.67,
NDFAIL,BOS,Demand,LINK18,LAX,,ATL,2488.32,2542.617,102.18,99.021,3.98,
NDFAIL,BOS,Demand,LINK2,ATL,,WDC,2488.32,3118.01,125.31,124.398,5.00,
NDFAIL,BOS,Demand,LINK13,SJC,,LAX,2488.32,3203.627,128.75,132.551,5.33,
NDFAIL,BOS,Demand,LINK14,PHI,,NYC,2488.32,2960.928,118.99,131.768,5.30,
NDFAIL,CHI,Demand,LINK18,LAX,,ATL,2488.32,3063.034,123.10,145.86,5.86,
NDFAIL,CHI,Demand,LINK2,ATL,,WDC,2488.32,3699.55,148.68,176.737,7.10,
NDFAIL,CHI,Demand,LINK2,WDC,,ATL,2488.32,2503.67,100.62,115.251,4.63,
NDFAIL,CHI,Demand,LINK13,SJC,,LAX,2488.32,3399.329,136.61,150.52,6.05,
NDFAIL,CHI,Demand,LINK14,NYC,,PHI,2488.32,2799.688,112.51,145.904,5.86,
NDFAIL,CHI,Demand,LINK14,PHI,,NYC,2488.32,3456.82,138.92,176.398,7.09,
NDFAIL,DAL,Demand,LINK18,LAX,,ATL,2488.32,2884.802,115.93,129.819,5.22,
NDFAIL,DAL,Demand,LINK2,ATL,,WDC,2488.32,3840.348,154.33,189.409,7.61,
NDFAIL,DAL,Demand,LINK8,CHI,,WDC,2488.32,2670.248,107.31,121.782,4.89,
NDFAIL,DAL,Demand,LINK13,SJC,,LAX,2488.32,3382.091,135.92,148.97,5.99,
NDFAIL,DAL,Demand,LINK14,PHI,,NYC,2488.32,2987.2,120.05,134.132,5.39,
```

PeakSimRoute.x Report

The Peak Simulation Route Report can be generated by running failure simulation from menu Simulation > Predefined Scenarios. This report provides information on demands or tunnels that failed to be rerouted during the failure simulation.

Here is a portion of a sample peak simulation route report:

```
************************************************
*  PEAK SIMULATION DEMAND/TUNNEL FAILURE INFO
************************************************
##  Software Release= 6.0.1, 64 bits,  Compilation Date= 20120418
##  Platform=i86pc, OS=SunOS 5.10
##  Report Date= 4/20/2012 10:45   Runcode=mpls-fish  User=wandl
# List Demands failed to be routed
SimType,SimEvent,Layer,PathName,From,To,ToIPaddr,Bandwidth((Mbit)),Priority,Dist,Delay(ms),path
string,MiscInfo,Type
NDFAIL,CHI,Tunnel,RBOSWDC,BOS,WDC,,10,2/2,,,BOS-DET-CHI-WDC,Path
Required,"R,A2Z,MASK=0000ffff,PR10(BOS-DET-CHI-WDC)",
NDFAIL,CHI,Tunnel,RWDCBOS,WDC,BOS,,15,2/2,,,WDC-CHI-DET-BOS,Path Required,"R,A2Z,PR(WDC-CHI-DET-BOS)",
NDFAIL,CHI,Tunnel,RHOUWDC,HOU,WDC,,5.0,2/2,,,HOU-DAL-CHI-WDC,Path Required,"R,A2Z,PR(HOU-DAL-CHI-WDC)",
NDFAIL,DAL,Tunnel,RATLCHI,ATL,CHI,,1.0,2/2,,,ATL-HOU-DAL-CHI,Path
Required,"R,A2Z,MASK=0000ffff,PR10(ATL-HOU-DAL-CHI)",
NDFAIL,DAL,Tunnel,RHOUWDC,HOU,WDC,,5.0,2/2,,,HOU-DAL-CHI-WDC,Path Required,"R,A2Z,PR(HOU-DAL-CHI-WDC)",
NDFAIL,DAL,Tunnel,RSJCCHI,SJC,CHI,,5.0,2/2,,,SJC-LAX-SDG-HOU-DAL-CHI,Path
Required,"R,A2Z,PR(SJC-LAX-SDG-HOU-DAL-CHI)",
NDFAIL,DET,Tunnel,RBOSWDC,BOS,WDC,,10,2/2,,,BOS-DET-CHI-WDC,Path
Required,"R,A2Z,MASK=0000ffff,PR10(BOS-DET-CHI-WDC)",
NDFAIL,DET,Tunnel,RWDCBOS,WDC,BOS,,15,2/2,,,WDC-CHI-DET-BOS,Path Required,"R,A2Z,PR(WDC-CHI-DET-BOS)",
NDFAIL,HOU,Tunnel,RATLCHI,ATL,CHI,,1.0,2/2,,,ATL-HOU-DAL-CHI,Path
Required,"R,A2Z,MASK=0000ffff,PR10(ATL-HOU-DAL-CHI)",
NDFAIL,HOU,Tunnel,RSJCCHI,SJC,CHI,,5.0,2/2,,,SJC-LAX-SDG-HOU-DAL-CHI,Path
Required,"R,A2Z,PR(SJC-LAX-SDG-HOU-DAL-CHI)",
NDFAIL,LAX,Tunnel,RSJCCHI,SJC,CHI,,5.0,2/2,,,SJC-LAX-SDG-HOU-DAL-CHI,Path
Required,"R,A2Z,PR(SJC-LAX-SDG-HOU-DAL-CHI)",
NDFAIL,SDG,Tunnel,RSJCCHI,SJC,CHI,,5.0,2/2,,,SJC-LAX-SDG-HOU-DAL-CHI,Path
Required,"R,A2Z,PR(SJC-LAX-SDG-HOU-DAL-CHI)",
```

Chapter 10

IP/MPLSView Scripting

The following chapter describes the text-based alternative for automating the creation of a network model and the generation of web reports. The text-based option allows for flexible integration with other third-party data sources.

Scripting Overview

The following programs are used to collect data, extract data to create a network model, and to open the network model to generate reports.

- **getipconf**: Extracts data from configuration, interface files, etc.to build a network model.
- **rdjpath**: Extracts data from tunnel_path, transit_tunnel files about a tunnel’s actual placement and status (up or down).
- **rtserver**: Opens the network model after which simulations can be run given a demand file (as desired), and reports can be generated

After running these programs, the output generated by these programs can be exported to HTML using the rpt2html utility:

- **rpt2html**: Converts IP/MPLSView-generated reports into HTML format for viewing in the IP/MPLSView Web Interface

In text mode, the automation of these processes can be done through the cron job. (In the Java client interface, some of these processes, such as getipconf and web reports are automated through the Task Manager.)

Extracting the Network: getipconf

The getipconf program is used to extract data from router CLI output such as configuration and interface data, in order to create a network model that can be loaded in by the IP/MPLSView software.

Type the command /u/wandl/bin/getipconf to see the following options. Some of the more typical options (in bold font below) are described below. For more advanced options, refer to the Router Guide, “Router Data Extraction” chapter.

```
usage: /u/wandl/bin/getipconf [-r runcode] [-t topfile] [-b bwconvfile] [-n
muxloc [-p nodeparam]] [-intf intfmap] [-vrf vrffile] [-vlan vlanfile] [-noBGP]
[-noVPN] [-noCE] [-printDup] [-vpnName] [-STM] [-i interfaceDir] [-snmp SNMPDir]
[-commentBW] [-ignore ipaddr] [-ospf ospfdatabase] [-atmbw] [-as asNameFile]
[-group groupFile] [-coord graphCoordFile] [-nbr neighborDir]
[-ignoreIPUnnumbered] [-secondary] [-LSPDir lspDir] [-cdp cdpDir] [-EXSW
EXSWdir] [-spec spec] [-dparam dparam] [-checkMedia] [-exIC] [-noASNodeLink]
[-delay delayFile] [-PECE PECEfile] [-routeInstance routeInstanceFile]
```

```
[-bgpGroupTable file] [-srvcType file] [-allPolicies] [-greTunnel] [-iptraf]
[-baseIntf baseIntf] [-deltaIntf deltaIntf] [-cat selected category for report]
[-router selected router for report] [-filter filter for report] [-user
username] [-IC ICmessageList file name] [-cosalias cosaliasFile] [-mgnt] [-dir
configDir] [-profile profile] [-dummyNode] [-srp srpTopoFile] [-vlandiscovery
vlanDir] [-probe probeFile] [ config1 config2 ... [-tn topofiles...]
```

Common Options

- **-dir configDir** or **[config1 config2 ...]** The former option is used to specify a directory containing the router configuration files. The latter option is an alternative option to specify the individual files in a list such as config 1 config 2… or *.cfg. This latter format supports regular expressions.
- **-r runcode**: The runcode is the file extension to use for the files created for the new network model
- **-t topfile**: The MPLS topology file can be used for MPLS-TE enabled networks to construct the topology without the entire configuration files, using the output of *"show mpls traf topology*" (Cisco) or "*show ted database extensive*" (Juniper)
- **-n muxloc**: If you have a previous muxloc file with geographical coordinates in it, specify it here to preserve the geographical coordinate information.
- **-p nodeparam**: The nodeparam file is used to specify additional node details such as the hardware type (e.g., Cisco, Juniper, ASNODE, ETHERNET), and it is recommended to specify it along with the muxloc file. Otherwise, if a node is in the muxloc file, and if the configuration file(s) do not have the data for this node, then the node will count as a normal node towards the node license limit, when it could actually be a pseudo-node such as ASNODE or Ethernet node which does not count towards the limit.
- **-i interfaceDir**: The interfaceDir is the directory that contains the "show interfaces" command line output and is used to obtain the bandwidth information, which is not always available in the configuration files, e.g., for POS links.
- **-ignore ipaddr**: This parameter is used to ignore certain subnets, such as private IP address subnets, to avoid stitching links based on subnet matching for these IP address ranges.
- **-delay delayFile**: Specifies a comma-separated file with the link delay information. Of the following fields, only the router, interface, and ZtoA latency in milliseconds are required. Without this file, the delay will be estimated based on airline mileage between the two endpoints, if geographical coordinates are provided.

Examples

```
/u/wandl/bin/getipconf -r x -n /export/home/wandl/network/muxloc.old -as
/u/wandl/db/misc/ASNames -noMedia -i /data/interfaces -dir /data/config
/u/wandl/bin/getipconf -r x -n /export/home/wandl/network/muxloc.old -as
/u/wandl/db/misc/ASNames -noMedia -i /data/interfaces /data/config/*.cfg.*
```

After running getipconf, the location of additional project files that are not generated by getipconf can be appended to the end of the **spec** file. For more information on the spec file format, refer to Chapter 2, Spec File. An absolute path can be specified for the path of the project file. Otherwise, the path is relative to the **datadir**, if specified.

```
echo "interfaceLoad_in=interface.in.auto" >> spec.auto
echo "interfaceLoad_out=interface.out.auto" >> spec.auto
echo "demand=demand.auto" >> spec.auto
```

Note that some of these project files can also be added to the spec file using an option in the getipconf script (e.g., -group and -coord for the group and graphcoord files, respectively). If those options are not available in the getipconf script, however, they can be manually added by appending them to the spec file as shown above.

Additional parameters can also be added into the **dparam** file as desired. For more information on the dparam parameters, refer to Chapter 3, DPARAM File. This file can be used to specify locale-specific parameters such as the distance unit (miles vs. kilometer). The field “batch=1” is helpful for rtserver scripting, which will be discussed next.

```
echo “batch=1” >> dparam.auto
echo “distunit=km” >> dparam.auto
```

Opening the Network and Creating Reports: rtserver/bbdsgn

The routing engine rtserver can be used to load the network model created by getipconf. This program can load in the demand file, route demands, run simulations, report on link capacity and demand paths, etc. Many of the functionality from the Java interface can be done in text mode through the rtserver program, via text menus. Rtserver can be invoked using the command /u/wandl/bin/bbdsgn, which is a link to the rtserver program.

The scripting of rtserver relies upon specification the keystrokes to various menus. To automate this process, an initial run needs to be performed to save the keystrokes.

1. Run the program with the additional parameter “traceinput” to save the keystrokes into the file input.trc:

```
$ /u/wandl/bin/bbdsgn /u/wandl/sample/IP/fish/spec.mpls-fish traceinput
```

2. For example, answer: y, y, 4, REPORT, 3, 2, q, 1, 2, q, q, y which will load in demands and generate some reports.
3. This creates the output file input.trc with these keystrokes, one per line. Rename the file input.trc so you do not overwrite it by accident in future runs.

```
$ mv input.trc input.report
```

4. Now rerun the program using these keystrokes.

```
$ /u/wandl/bin/bbdsgn /u/wandl/sample/IP/fish/spec.mpls-fish < input.report
```

The reports will then be created in batch. You can go through process 1-3, specifying additional reports to generate. You can even run a failure simulation and then generate simulation reports. The following are some Reports of interest available from the report menu:

- Path Report (PATHRPT)
- Link Utilization report (LKUTIL)
- Node Traffic Summary Report (In Report Menu, T., 1)
- Simulation Report: Run a Node and Link Failure Simulation and report on the peak (8, 4, 2, a, 1, etc.)

If you are designing an input file for a large network, it is better to test your script first on a smaller example of the same hardware vendor (e.g., do not use a script for an ATM network on a router network.)

Informational Note: The program can be slightly non-deterministic, with the questions varying slightly depending upon the network, or the routing of demands, etc. Hence, every time you get back to the main menu, it is recommend to put a couple of extra new lines in case of an extra unexpected question (an empty line is equivalent to pressing enter to take the default answer.) Sometimes, it is also helpful to document the keystrokes by putting a # sign after the response, although with care because the # sign is not always ignored. The following is an example keystroke file.

```
n # Route Unplaced Tunnels
n # Auto Bypass Tunnels
n # Update demand routing tables (saying yes can be nondeterministic here, e.g.,
questions can appear or not appear depending upon the number of unplaced
demands, so the demand rerouting will be done manually instead in the next set
of lines)


6 # Path Placement
2 # Modify Path Placement
2 # Design from Scratch
1 # Sequential
n # Interactive
1 # All paths
n # Verbose
q # quit
q # quit to main menu


4 # Reports
RPT
3 # Report Style
2 # CSV
q # Quit to Report Menu
2 # Link Bandwidth Allocations (saves LKUTIL.RPT, utilcolor.RPT, and
LKSRVCUTIL.RPT)
T # Traffic Statistics Report
1 # Node Traffic Summary Report SWITCHCONN.RPT
q # Quit
q # brings back to main menu



q # Exit main menu
y # Exit BBDSGN
```

The reports generated by this program can be converted into HTML using rpt2html.

Importing Tunnel Paths: rdjpath

The following command is used to extract tunnel path data. See, the Router Guide chapter, “Router Data Extraction” for details of the tunnel path data format.

```
/u/wandl/bin/rdjpath -r runcode tunnel_path_dir
```

Substitute the *runcode* with the same file extension used by your network project and *tunnel_path_dir* with the directory containing the tunnel path files collected from the router.

The resulting file, tunnelpath.*runcode* can be imported into the network via /u/wandl/bin/bbdsgn, option M. MPLSView > 3. Read MPLS Tunnel Path. This can also be automated via the input trace file.

Creating Hardware Inventory Reports

The Hardware Inventory module uses a Java program which requires specifying Java library files. The prerequisite is to first collect equipment_cli data, which can be done using IP/MPLSView.

```
WANDL_HOME=/u/wandl
hardwaredir=
outdir=
specfilepath=
/u/wandl/java/bin/java -cp
$WANDL_HOME/lib/wandl/bki.jar:$WANDL_HOME/lib/thirdparty/xerces.jar:$WANDL_HOME/
lib/wandl/tmng.jar com.wandl.generic.inventory.Console -sync -cli $hardwaredir
-v -o $outdir $specfilepath
```

Specify the values of the following variables:

- $hardwaredir: Directory containing the collected hardware information
- $outdir : A Directory to save the report outputs of this program in CSV format. The reports generated here can then be converted into HTML using rpt2html.
- $specfilepath: The absolute path of the spec file

If desirable, you can create a different outdir for each day by including the date in the directory name, e.g., using `date +%d%m%Y` as part of the outdir in shell script.

Converting Reports to HTML: rpt2html

The rpt2html utility is used to convert CSV reports to HTML for display on the IP/MPLSView Web Interface. Type /u/wandl/bin/rpt2html for the usage information:

```
Usage: rpt2html -s specfile [-o source report dir] [-r runcode] [-h] [-l all | list of report numbers
separated by ','] [-t auto | last | date separated by '_']
-h: hide the "Layer 3" in the converted html report filename
```

After running getipconf and rtserver or the hardware inventory report script to create the types of reports you want, you can run rpt2html to convert specific reports to HTML. The following is an example of an rpt2html command which will convert all available reports for a specific network with file extension “x” to HTML.

```
/u/wandl/bin/rpt2html -s /export/home/wandl/baseline/spec.x -r x -l all -t auto
```

Note that for reports in Report > Report Manager, you can also directly invoke report generation and html conversion via the following bbdsgn (text mode) options: 4. Reports > Enter runcode > quit > W. Save all report and export to the Web.

Following this, you can open up the IP/MPLSView Web Interface > Network Reports > Select Spec Path > Select Date to see the available web reports. The reports will be generated under /u/wandl/data/.WEB.

Following are further instructions for the rpt2html parameters:

- **-s specfile:** It’s recommended to specify the full path of the spec file here.
- **[-o source report dir]:** This option is required if the output CSV files are not in the same directory as the spec file. For example, this may happen when processing the hardware equipment inventory reports into a separate folder.
- **[-r runcode]:** This is the file extension of the CSV files that is required for the conversion of those report to HTML. Note that even if the spec file is spec.x, the report could have been created using a different file extension, such as .x.csv, or whatever arbitrary file extension the user chose.
- **[-l all | list of report numbers separated by ‘,’]:** If selecting “all” then all the reports that are discovered for the given spec file in the given source report dir, if specified, will be converted. A caveat about rpt2html is that it only converts existing reports into HTML. It does not re-generate the report itself, so it is possible for the report to contain outdated information if it was not recently regenerated through getipconf, rtserver, or the hardware equipment script. Hence, instead of generating all reports using the “-l all” option, it is therefore a good idea to selectively create HTML reports for the reports that were recently generated. For example, if the script had recently run getipconf, then the Integrity Checks Report (e.g., configLog.auto) could be converted to HTML using the report code 19 (-l 19).
- **[-t auto | last | date separated by '_']:** When reports are generated to the web, a date must be specified. When selecting “-t auto”, the date will be auto-generated based on the current time. When selecting “-t last”, the report will be merged with the last generated time for that spec file. The final option is to specify a specific date, e.g., December_19_2007_14_44_28.

```
1. Example for converting all reports with a particular time stamp:
/u/wandl/bin/rpt2html -s /u/wandl/sample/IP/fish/spec.mpls-fish -r mpls-fish -l all -t
December_19_2008_3_30_56
2. Example for converting only the PATHRPT (Path Report) and LKUTIL (Link Utilization) report using an
auto-generated date for the current time:
/u/wandl/bin/rpt2html -s /export/home/wandl/baseline/spec.x -r x -l 1,11 -t auto
3. Example for converting only hardware reports from a specified output directory, and merging it with
the last time stamp generated for this spec project:
/u/wandl/bin/rpt2html -s /export/home/wandl/baseline/spec.x -o /export/home/wandl/baseline/hwequip -r
auto.csv -h -l 2018,2019,2020,2021,2022,2023,2024,2025 -t last
```

Another alternative to rpt2html is to use the Task Manager, Web Report task.

Automating the Scripts on a Regular Basis: Crontab

The final step in the automation in text mode is to create a cron job. Note that cron jobs need to use the full path for the command as well as the arguments to those commands. If the cron job is used to run a script, absolute paths should ideally be specified in the script as well. The following is the Crontab format:

```
min(0-59) hour(0-23) day(1-31) month(1-12) day_of_week(0-6,Sunday=0) command
```

In later releases of Solaris, like Solaris 10, the cron job can be run under different users. Login as the user that will be running the cron. Then use the following commands to view and edit the cron.

```
export EDITOR=vi        set the editor with which to edit the cron job, e.g.
using vi
crontab -l              view the cron
crontab -e              edit cron
```

One shell script can be created to run all necessary processes, e.g., /export/home/wandl/my_cron.sh can be created to call a) getipconf b) bbdsgn c) Java for hardware inventory, and d) rpt2html. Specify one command per line, and specify absolute paths for the cron job to be safe (e.g., “/u/wandl/bin/bbdsgn /export/home/wandl/spec.xxx < /export/home/wandl/input.report” rather than “bbdsgn spec.x < input.report”). Alternatively, the shell script can be split into multiple jobs, but this requires more timing coordination to ensure that the prerequisites are done before they are needed.

Make the shell script executable:

```
chmod +x my_cron.sh
```

Then add it to the cron tab. The cron job below would mean: Run my script everyday at 7:30am.

```
30  7 * * * /export/home/wandl/my_cron.sh
```

Reporting Codes for rpt2html

The following are the report codes that can be used for rpt2html. Note that some of these reports are hardware and license dependent. The utility rpt2html will only convert files that are already generated.

Report Description	Report File Name	Report ID
PathandDiversityReport	PATHRPT	1
TunnelPathandDiversityReport	TUNNELRPT	1001
BandwidthAllocationReport	LKBWRPT	2
BandwidthAllocationReport(Layer2)	L2_LKBWRPT	3002
LinkCostReport	LKCOST	3
Link(Tunnel)PartitionInformation	LKPART	4
EquivalentCapacityAllocationInformation	LKEQCAP	1004
HardwareCostReport	BBHWCRPT	5
RouteCostReport	CKTCOST	6
RouteCostReport(Layer2)	L2_CKTCOST	3006
InterDomainLoadDistributionReport	INTDOMLOAD	8
InterDomainPathsReport	INTDOMPATH	9
Domain(Area)PassthroughPathsReport	DOMPASSTHRU	10
LinkUtilizationReport	LKUTIL	11
LinkUtilizationReport(Layer2)	L2_LKUTIL	3011
LinkLoadReport	LINKLOAD	12
EquivalentCapacityReport	PATHBW	13
EqualCostMulti-PathReport	EQPATHRPT	14
ChannelAssignmentReport	CHANTBL	15
CoSDemandsReport	DEMANDCOS	16
MeasuredTunnelTrafficReport	TUNNEL_LOAD	1016
CoSLinksReport	LINKCOS	17
CoSLinksReport(Layer2)	L2_LINKCOS	3017
RoutingTableReport	RTTABLE	18

Report Description	Report File Name	Report ID
ConfigurationIntegrityChecksReport	configLog	19
BGPIntegrityCheckReport	BGPRPT	20
InternationalCostReport	INTLCOST	21
CountryCostReport	CTRYCOST	1021
VoiceSummaryReport	VOICERPT	22
VoiceTrunkGroupReport	TKGPRPT	23
VoiceRoutingTableReport	VRTRPT	24
VoicePathReport	VPATHRPT	25
SwitchConnectionsStatisticsReport	SWITCHCONN	26
TunnelStatisticsReport	TUNNELSTAT	1026
HPNNIIntegrityReport	HPNNIIntegrity	27
HPNNIPeerGroupStatisticsReport	HPNNISTAT	28
RerouteStatisticsReport	REROUTESTAT	29
NodeInventoryReport	NODEINV	30
CPSSDomainStatisticsReport	CPSSDmnStat	31
CPSSIntegrityCheckReport	CPSSIntChk	32
LinkConfigReport	linkconf	33
SwitchPNNIConfigurationReport	SWITCH_PNNI_CONFIG_RPT	34
LinkPeakUtilizationReport	PEAKUTIL	35
LinkPeakUtilizationReport(Layer2)	L2_PEAKUTIL	3035
OSPFAreaSummaryReport	OSPFSUMMARYRPT	36
ABRBorderingAreaReport	ABRBORDERRPT	37
OSPFAreaDetailReport	OSPFDETAILRPT	38
InterfaceTrafficReport	IntfUtil	39
TunnelTrafficReport	TUTRAFRPT	40
ProvisionedTunnelUtilization(ConfiguredTunnelBandwidth)Report	TUTIL	41
NodeProcessorStatisticsReport	NODE_PROCLOAD	42
LinkProcessorStatisticsReport	LINK_PROCLOAD	43
Layer3VPNReport2	VPNREPORT_LAYER3	44
Layer2VPNReport	VPNREPORT_LAYER2	45
ASTrafficReport	ASTraffic	46
Inter-ASTrafficReport	InterASTraffic	47
ISISReport	ISISReport	48
GroupLinkSUMMARYReport	GROUPLINKSUMMARY	101
GroupLinkDETAILReport	GROUPLINKDETAIL	102
GroupDemandSUMMARYReport	GROUPDEMANDSUMMARY	103
GroupDemandDETAILReport	GROUPDEMANDDETAIL	104
GroupInterfaceSUMMARYReport	GROUPINTFSUMMARY	105
GroupInterfaceDETAILReport	GROUPINTFDETAIL	106
InteractiveFailureReport	SIMRPT	108
ExhaustiveNodeFailureReport	NDFAIL	109

Report Description	Report File Name	Report ID
ExhaustiveNodeFailureReport(Layer2)	L2_NDFAIL	3109
ExhaustiveSingleLineFailureReport	LINEFAIL	110
ExhaustiveSingleLineFailureReport(Layer2)	L2_LINEFAIL	3110
ExhaustiveFacilityFailureReport	FACFAIL	111
ExhaustiveFacilityFailureReport(Layer2)	L2_FACFAIL	3111
ExhaustiveSiteFailureReport	SITEFAIL	112
ExhaustiveSiteFailureReport(Layer2)	L2SITEFAIL	3112
ExhaustiveLinkFailureReport	LKFAIL	113
ExhaustiveLinkFailureReport(Layer2)	L2_LKFAIL	3113
RandomDailyFailureReport	DAILYFAIL	114
RandomDailyFailureReport(Layer2)	L2_DAILYFAIL	3114
DailySequenceReport	DAILYSEQ	115
DailySequenceReport(Layer2)	L2_DAILYSEQ	3115
LinkDiversityUtilizationReport	DVSIM	116
LinkDiversityUtilizationReport(Layer2)	L2_DVSIM	3116
PathDelayInformationReport	PATHDELAY	117
PathDelayInformationReport(Layer2)	L2_PATHDELAY	3117
Up-DownSequenceReport	UPDOWN	118
Up-DownSequenceReport(Layer2)	L2_UPDOWN	3118
DiscreteEventReport	EVENTTRC	119
VoiceTrafficSummaryReport	VTRAFRPT	120
LinkRuleVerificationReport	LKDSGNRULERPT	121
IncrementalPartitionDesignReport	PARTDSGN	122
SummaryofIntegrityChecksReport	Integrity_Summary_rpt	123
OSPFReport	OSPFReport	
GroupTunnelSUMMARYReport	GROUPTUNNELSUMMARY	129
GroupTunnelDETAILReport	GROUPTUNNELDETAIL	130
ExhaustiveCardFailureReport	CARDFAIL	131
ExhaustiveCardFailureReport(Layer2)	L2_CARDFAIL	3131
ExhaustiveVoiceTrunkGroupFailureReport	VTKGPFAIL	132
ExhaustiveVoiceTrunkGroupFailureReport(Layer2)	L2_VTKGPFAIL	3132
ReplayUp-DownSequenceReport	REPFAIL	133
ReplayUp-DownSequenceReport(Layer2)	L2_REPFAIL	3133
SharedRiskLinkGroupsReport	SRLG	134
SharedRiskLinkGroupsReport(Layer2)	L2_SRLG	3134
CustomerReport	CUSTOMREPORT	2000
ReplayFailureSimulationReport	SIMFAIL	135
PathPlacementSimulationReport	SIMPLACE	136
SystemLimitOverAllReport	SYSTEMLIMITOVERALL	137
SystemLimitExceed(TCA)Report	SYSTEMLIMITEXCEEDTCA	138
VPNInterfaceTrafficReport	VPNINTFTRAF	139
VPNExportImportReport	VPNEXPORTIMPORT	140

Report Description	Report File Name	Report ID
VoIPCallSetupReport	VOIPCALLSETUPRPT	141
VoIPNodeTrafficSummaryReport	VOIPSWITCHCONN	145
VoIPNodeTunnelSummaryReport	VOIPTUNNELSTAT	2145
VPNLayer2KompellaReport	VPNREPORT_L2KOMPELLA	146
VPNLayer2CCCReport	VPNREPORT_L2CCC	147
VPNVPLSReport	VPNREPORT_VPLS	148
VPNLayer2MartiniReport	VPNREPORT_L2MARTINI	149
VPNVPLS(Juniper)Report	VPNREPORT_VPLS_JUNIPER	150
SubnetReport	SUBNETRPT	151
ProvisionedNodeLoadReport	PROVNODELOAD	152
MeasuredNodeLoadReport	MEASNODELOAD	153
LkUtilizationReport(Layer3)	L3LINKUTIL	154
MeasuredLinkUtilizationReport(Layer2)	L2_MEASUREDLKUTIL	155
MeasuredLinkUtilizationReport(Layer2)	L2MEASUREDLKUTIL	1155
In/OutboundNetworkTrafficReport	NWKTRAF	156
CoSReport	CoSReport	157
ComparedLinkUtil(DemandvsIntf)Report	L3LINKUTILCOMPARE	158
ComparedLinkUtil(TunnelvsIntf)Report	LINKUTILCOMPARE	159
VoiceNodeBlockingSummaryStatisticsReport	VNODETRAFRPT	160
autotunneldiscrepancyReport	ATDISCREPANCYRPT	161
autotunnelprotectionReport	ATPROTECTIONRPT	162
L2GroupTrafficSummaryReport	GROUPLOAD	163
L3GroupTrafficSummaryReport	L3GROUPLOAD	164
L3GroupedLink(Measured)Report	MEASUREDLKBWRPT	165
L2GroupedLink(Measured)Report	L2_MEASUREDLKBWRPT	1165
VoIPEModelReport	VOIPEMODEL	166
RouteInstanceReport	RTINSTRPT	167
Layer3VPNReport	L3VPNReport	168
ExhaustiveSlotFailureReport	SLOTFAIL	169
TunnelPeakUtilReport	TUNNELPEAKUTIL	170
NodemanagerReport	NODEMANAGER	171
LinkmanagerReport	LINKMANAGER	172
PerServiceReport	LKSRVCUTIL	173
CompareLinkUtil(After-Before)Report	PEER_LINKUTIL	174
CompareNodeTraffic(After-BeforeReport	PEER_NODETRAF	175
ProvisionedELSPQoSReport	ELSPQOS	176
MeasuredELSPQoSReport	MEASURED_ELSPQOS	177
MeasuredLinkUtilizationReport(basedoninterface)	MEASURED_LINKUTIL_INTF	178
AutoTUNNELOverlapReport	ATOVERLAPRPT	179
VLANDetailReport	VLANReport	180
RNDPathReport	RNDPATH	2011
PathReport	PATH	2012

Report Description	Report File Name	Report ID
LinkUtilizationReport(RNDLKUTL)	RNDLKUTL	2013
LinkUtilizationCompareReport	CMPLKUTIL	2015
RouteCostReport	CKTCOST_RT	2016
RouteCostReport(Layer2)	L2_CKTCOST_RT	4016
LinkDiversityUtilizationReport	DVUTIL	2017
LinkDiversityUtilizationReport(Layer2)	L2_DVUTIL	4017
RoutersList	ROUTER_LIST	2018
CardsList	CARD_LIST	2019
PhysicalInterface	INTF_LIST	2020
HardwareIntegrity	HW_IC_REPORT	2021
DeviceUsage	HW_DEVICE_USAGE_REPORT	2022
LineCardUsage	HW_LINECARD_USAGE_REPORT	2023
HardwareCapEx	HW_CAPEX_REPORT	2024
CategorizedbyParts	HW_PARTS_REPORT	2025
CustomizedReports	customized_	9999